

UNIVERSIDADE FEDERAL DO PARANÁ

LUIS CARLOS FERREIRA BUENO

CONCEITOS E FERRAMENTAS DE AUTORIA PARA REPRESENTAÇÃO DA
VISÃO ESTRATÉGICA BASEADA EM PADRÕES E
BUSCA EM JOGOS HEURÍSTICOS ADVERSARISTAS

CURITIBA PR

2018

LUIS CARLOS FERREIRA BUENO

CONCEITOS E FERRAMENTAS DE AUTORIA PARA REPRESENTAÇÃO DA
VISÃO ESTRATÉGICA BASEADA EM PADRÕES E
BUSCA EM JOGOS HEURÍSTICOS ADVERSARISTAS

Tese apresentada como requisito parcial à obtenção do grau de Doutor em Ciência da Computação no Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação*.

Orientador: Marcos Castilho, Dr..

Coorientador: Bruno Muller Junior, Dr..

CURITIBA PR

2018

Catálogo na Fonte: Sistema de Bibliotecas, UFPR
Biblioteca de Ciência e Tecnologia

B928c Bueno, Luis Carlos Ferreira

Conceitos e ferramentas de autoria para representação da visão estratégica baseada em padrões e busca em jogos heurísticos adversaristas [recurso eletrônico] / Luis Carlos Ferreira Bueno – Curitiba, 2018.

Tese (doutorado) - Programa de Pós-graduação em Informática, Setor de Ciências Exatas, da Universidade Federal do Paraná.

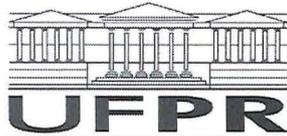
Orientador: Prof. Dr. Marcos Castilho

Coorientador: Pro. Dr. Bruno Muller Junior

1. Inteligência artificial. 2. Reconhecimento de padrões. 3. Jogos educacionais. I. Universidade Federal do Paraná. II. Castilho, Marcos. III. Muller Junior, Bruno. IV. Título.

CDD 006.3

Bibliotecária: Vilma Machado CRB9/1563



MINISTÉRIO DA EDUCAÇÃO
SETOR SETOR DE CIÊNCIAS EXATAS
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO INFORMÁTICA -
40001016034P5

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em INFORMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da tese de Doutorado de **LUIS CARLOS FERREIRA BUENO** intitulada: **Conceitos e Ferramentas de Autoria para Representação da Visão Estratégica Baseada em Padrões e Busca em Jogos Heurísticos Adversaristas.**, após terem inquirido o aluno e realizado a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de doutor está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 11 de Fevereiro de 2019.


MARCOS ALEXANDRE CASTILHO
Presidente da Banca Examinadora


ANDREY RICARDO PIMENTEL
Avaliador Interno (UFPR)


ALEXANDRE RÔMULO MOREIRA FEITOSA
Avaliador Externo (UTFPR)


WILSON DA SILVA
Avaliador Externo (UNINTER)


FABIANO SILVA
Avaliador Interno (UFPR)



Este trabalho é dedicado aos entusiastas dos estudos em Inteligência Artificial e aos exadristas cujo conhecimento foi e tem sido usado como campo de testes para muitos avanços científicos. Também dedico este trabalho ao grande Professor Alexandre I. Direne que deixou saudades.

Agradecimentos

À Deus e pela maravilhosa pessoa de seu Filho Jesus pela presença constante nas horas mais amargas.

Aos meus pais Oscar Ferreira Bueno (in memoriam) e Angelina Slusarski (in memoriam) por tudo que investiram em minha vida e pelo exemplo de força e dedicação.

À minha esposa Beatriz P. Bueno, mulher de fibra e guerreira. Sem ela não chegaria tão longe. Obrigado pela paciência, amor, cobrança e incentivo.

À minha filha Laura, por compreender-me nos momentos de estresse e ausência.

Ao meu irmão Paulo pelo apoio e incentivo nas horas de medo.

Ao grande professor e orientador Alexandre I. Direne (in memoriam) por ter me dado suporte, incentivo e pelo seu esforço em responder minhas indagações mesmo em condições de saúde debilitada na fase inicial do trabalho.

Ao Prof. Bruno Muller, cujo conhecimento enxadrístico e computacional, me ajudou a chegar aqui. Pela sua compreensão nas dificuldades de ordem pessoal e pela presteza em me atender e redirecionar o trabalho em momentos de dificuldade.

Ao Prof. Dr. Wilson Silva e membros do Clube de Xadrez Erbo Stenzel que me atenderam em alguns momentos durante a fase de qualificação e definições do trabalho.

Aos membros da banca, professores: Dr. Alexandre Feitosa, Dr. Andrey Pimentel, Dr. Fabiano Silva e Dr. Wilson Silva pelas críticas construtivas que certamente contribuíram em muito com o trabalho.

Aos demais amigos e colegas que de alguma forma contribuíram com este trabalho, em especial à Professora Dra. Beatriz T. Borsoi.

E por último, mas não menos importante, aos meus irmãos de fé que oraram por mim durante os tempos de ansiedade e problemas de saúde.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

RESUMO

A presente tese apresenta novos conceitos e ferramentas de *software* para dar suporte à formalização de conhecimento no nível de visão estratégica. No Xadrez, por exemplo, é a visão estratégica que permite ao enxadrista identificar quando duas posições de jogo diferentes apresentam as mesmas características, isto é, o mesmo padrão. A autoria formal de padrões de Xadrez é apresentada como um elemento chave sob o qual reside o estilo humano de jogar e como uma tarefa metacognitiva para aprendizes e grandes mestres refletirem sobre padrões armazenados em suas memórias de longo prazo. Conceitos de múltiplas representações externas são usados no projeto do ambiente. Poucos trabalhos foram encontrados na literatura científica sobre aplicar conhecimento, no nível de visão estratégica, para máquinas de jogo. Uma ontologia para especificar padrões é definida, bem como, um *ambiente* e uma linguagem para formalizar o conhecimento sobre padrões são apresentados. Experimentos para verificar o poder de expressividade foram realizados cujos resultados são promissores. Os conceitos abordados aqui podem ser utilizados em outras áreas como medicina e ciências administrativas.

Palavras-chave: inteligência artificial, IA, reconhecimento de padrões, xadrez, jogos educacionais, representação de conhecimento, formalização do conhecimento, visão estratégica.

ABSTRACT

This thesis presents new concepts and software tools to support knowledge formalization at the strategic vision level. In Chess, for example, is the strategic vision that allows chess players identifying when two different game positions has the same characteristics, it means, the same pattern. The formal authoring of patterns in Chess is presented as key element that underlies human-like style of playing and as a meta-cognitive task for learners and grand masters to reflect about patterns stored in their long term memory. Multiple external representation concepts are use in the design of the environment. Few past works have been found in the scientific literature about applying knowledge, at a strategical vision level, to game engines. An ontology for pattern specification has been defined, as well, an environment and a language for pattern formalization are presented. Experiments were done to verify the expression power of the language giving promising results. The concepts approached here can be used in other scientific fields as medicine and administration.

Keywords: artificial intelligence, AI, pattern recognition, chess, educational games, knowledge representation, knowledge formalization, strategic vision.

Lista de Figuras

1.1	Similaridade em nível de estratégia - Fonte: adaptado de [40]	18
1.2	Movimento tático x estratégico - brancas jogam - Fonte: adaptado de [43] .	21
2.1	Treinamento usando o VR-Simulator	35
2.2	UMRAO - Codificação de planos	37
2.3	Combinação de peças encontrada para aplicação de planos	42
2.4	Posição analisada - brancas jogam	43
2.5	Representação de padrões dinâmicos - Fonte: [24]	45
2.6	Estudo de caso - Fonte: [78]	45
2.7	Plano otimista genérico - Fonte: [78]	45
3.1	Ontologia de HLPs - Diagrama de classes	50
3.2	Ontologia de HLPs - Relacionamentos	51
3.3	Arcabouço do <i>StratVision</i>	54
4.1	Exemplos de instâncias de ataque duplo estático	58
4.2	Exemplo de instância de ataque duplo dinâmico	58
4.3	Exemplos de instância de ataque a descoberto	59
4.4	Exemplos de instância de ataque raio-x	59
4.5	Ferramenta de autoria PATLAN - <i>StratVision</i>	62
4.6	Saída do processo de busca de padrões - <i>StratVision</i>	64
5.1	Exemplo de execução do Philidor - Fonte: adaptado de [40]	68
5.2	Estudo de caso - Fonte: [40]	69
5.3	Estudo de caso - Fonte: Adaptado de [40]	69
5.4	Padrão Philidor encontrado - Fonte: [40]	71
5.5	Padrão Philidor encontrado - Fonte: adaptado de [40]	71
5.6	Interface para busca em bases pgn	72
5.7	Posições de convergência encontradas pelo StratVision - Brancas - Experimento 2	74
5.8	Posições de convergência encontradas pelo StratVision - Pretas - Experimento 2	75
5.9	Cenários similares encontrados pelo StratVision - Experimento 2	75
5.10	Verdadeiros positivos - Experimento 3	77
5.11	Falsos positivos - Experimento 3	78
5.12	Cenário Anastasia - falso positivo amigo	80
5.13	Cenário Anastasia - falso positivo interposto	81
5.14	Posições encontradas - Experimento 3	86
5.15	Exemplo de HLP passivo - Fonte:[40]	87
5.16	Exemplo de posições positivas encontradas - HLPs passivos - Brancas - Experimento 2	90

5.17 Exemplo de posições positivas encontradas - HLPs passivos - Pretas -	
Experimento 2	90
5.18 Falso positivo encontrado - HLPs passivos - Experimento 2	90

Lista de Tabelas

1.1	Avaliação estática - Posições Philidor	22
2.1	Linguagem para construção de planos - Pitrat	41
3.1	Relação de propriedades objeto da Ontologia	52
3.2	Relação de propriedades de dados da classe HLP	52
3.3	Instâncias por classe	52
4.1	Notação PATLAN	57
4.2	Exemplos de funções pré-definidas da linguagem PATLAN	60
5.1	Resultados busca Philidor - chessgames.com	73
5.2	Cenários similares ao Philidor encontrados	73
5.3	Resultados - Busca na base chess-db	76
5.4	Posições positivas - experimento 3 - Padrão Philidor	76
5.5	Falsos positivos - experimento 3 - Padrão Philidor	78
5.6	Explicação detalhada da listagem 5.4	83
5.7	Explicação detalhada da listagem 5.5	84
5.8	Resultados da busca - Anastasia (Vencedores em negrito)	85
5.9	Resultados da busca - Anastasia - Cenário 01	86
5.10	Resultados da busca - Anastasia - Cenário 02	87
5.11	Resultados da busca - Anastasia - Cenário 02	88
5.12	Verdadeiros positivos - experimento 2	89
5.13	Falsos positivos - experimento 2	89

Lista de acrônimos

BNF	Backus-Naur-Form
CACAREJE	Colaboração Alternada com Competição na Aprendizagem Referenciada por Jogos Educativos
CAI	Computer Assisted Instruction
CEX	Centro de Excelência de Xadrez
CGDH	Comparador Genérico de Definições Heurísticas
COA	Course of Actions
DHJOG	Definições Heurísticas de Jogos
EBL	Explanation-Based Learning
ENIAC	Electronic Numerical Integrator Analyzer and Computer
FAE	Função de Avaliação Estática
FIDE	Federação Internacional de Xadrez
GIFT	Generalized Intelligent Framework for Tutoring
HLP	High Level Pattern
I-CAI	Intelligent Computer Assisted Instruction
IA	Inteligência Artificial
IAC	Instrução Auxiliada por Computador
ICONCHESS	INteractive CONsultant for CHESS Middlegames
LEBL	Lazy Explanation-Based Learning
MCTS	Monte-Carlo Tree Search
MRE	Múltiplas Representações Externas
PARADISE	Pattern Recognition Applied to DIrecting SEarch
PATLAN	Linguagem para representação/programação de padrões
PROTEX	Projeto de Tipificação do Ensino de Xadrez nas Escolas Brasileiras
PV	Principal Variation
RC	Representação de Conhecimento
STI	Sistemas Tutores Inteligentes
SWOT	Strengths, Weaknesses, Opportunities, Threats

Sumário

1	Introdução	14
1.1	Contextualização	14
1.2	Caracterização do problema	16
1.2.1	Definição	16
1.2.2	Problemas de busca e algoritmos de busca heurística	19
1.3	Representação de conhecimento	22
1.3.1	Formalização do conhecimento heurístico	23
1.4	Justificativa	25
1.5	Hipóteses de pesquisa	26
1.6	Objetivos	26
1.6.1	Objetivo geral	27
1.6.2	Objetivos específicos	27
1.7	Considerações finais	27
2	Revisão de Literatura	28
2.1	Informática na Educação	28
2.1.1	Breve história	28
2.1.2	Avanços e desafios	30
2.1.3	Múltiplas Representações Externas	32
2.2	Aprendizado Baseado em Jogos Digitais	33
2.2.1	O problema da motivação	33
2.2.2	Aquisição de Conceitos e Habilidades	34
2.3	Xadrez, Informática e Educação	36
2.3.1	Trabalhos correlatos	36
2.3.2	IA aplicada a jogos de tabuleiro	37
2.4	Estratégia e ferramentas de planejamento	39
2.5	Jogadores automáticos e planejamento estratégico	40
2.6	Trabalhos recentes	46
3	Materiais e métodos	48
3.1	Tecnologias usadas	48
3.2	Elementos da Solução	49
3.2.1	Uma Ontologia para Representação de HLPs	50
3.2.2	Linguagem para Representação de HLPs	51
3.2.3	Ferramentas de Apoio	53
3.2.4	Experimentos	53
4	StratVision - Um Ambiente para Formalização da Visão Estratégica	55
4.1	PATLAN	55

4.2	Componentes de Software do Ambiente StratVision	61
4.2.1	Ferramenta de autoria	61
4.2.2	Gerador de Fatos	62
4.2.3	<i>Parser</i>	63
4.2.4	Busca de padrões - módulo de reconhecimento	65
4.2.5	Avaliação das instâncias	65
4.3	Resumo	66
5	Experimentos	67
5.1	Representação de HLPs ativos	67
5.1.1	HLP com sequência forçada de lances - Xequemate Philidor	68
5.1.2	Sequencia não forçada - Xequemate Anastasia	80
5.2	Representação de HLPs passivos	85
5.2.1	HLP - Rei pressionado por peão passado	86
6	Considerações Finais	91
6.1	Discussão	91
6.2	Impactos esperados	92
6.3	Conclusão	93
6.3.1	Limitações	94
6.3.2	Trabalhos futuros	94
	Referências	96
	Apêndice A: Definição da Linguagem de Padrões em Javacc	103
	Apêndice B: Tutorial da Linguagem PATLAN	113
	Apêndice C: Exemplo de Base de Fatos Prolog	127
	Apêndice D: Outros HLPs	140

1 Introdução

Este capítulo apresenta uma visão geral da presente tese em relação ao seu contexto, problema e objetivos, e encontra-se assim dividido: a seção 1.1 apresenta o contexto onde este trabalho está inserido. A seção 1.2 apresenta o problema que está sendo tratado. A seção 1.3 aborda o tema central dentro da Inteligência Artificial em que este trabalho está situado. A seção 1.4 apresenta uma justificativa do porquê é importante resolver o problema abordado. A seção 1.5 apresenta as hipóteses levantadas sobre o problema abordado. A seção 1.6 relaciona os objetivos a serem alcançados e, finalmente, a seção 1.7 apresenta considerações relevantes sobre esta tese.

1.1 Contextualização

A presente tese insere-se no Projeto de Tipificação do Ensino de Xadrez nas Escolas Brasileiras (PROTEX), apresentado em [21] e teve o apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001. O PROTEX está integrado ao Centro de Excelência de Xadrez (CEX), uma iniciativa do governo do Estado do Paraná que prevê a construção de ferramentas que possam ser usadas por alunos das escolas públicas brasileiras. Dentre os trabalhos já desenvolvidos neste projeto encontram-se os realizados por Hartmann [27], Martineschen *et. al.* [45], Feitosa [23] e Bueno [11].

Hartmann apresenta uma ferramenta de autoria para ensino de xadrez que auxilia na busca de características de harmonia de peças similares entre partidas. Permite aos professores produzirem material didático para ensino sobre o conceito de harmonia de peças auxiliados por um motor inteligente baseado em uma rede neural tipo Kohonen¹.

Martineschen apresenta a modelagem do ambiente de competição e colaboração, denominado como Colaboração Alternada com Competição na Aprendizagem Referenciada por Jogos Educativos (CACAREJE), e um protótipo de ferramenta para comparação de heurísticas adaptado para o jogo de Mancalla, denominado de Comparador Genérico de Definições Heurísticas (CGDH).

Feitosa descreve uma ferramenta de autoria, denominada de HeuChess, que permite aos aprendizes de xadrez definirem e inscreverem heurísticas de avaliação de tabuleiro em uma competição simulada. Apresenta também uma linguagem para formalização de heurísticas denominada de Definições Heurísticas de Jogos (DHJOG) .

Para fazer uso desta linguagem é necessário a instalação da ferramenta (desenvolvida em Java) que implementa os conceitos do ambiente CACAREJE. Esta ferramenta permite que aprendizes e tutores formalizem suas heurísticas através da DHJOG. Mais

¹O algoritmo de Kohonen foi desenvolvido por Teuvo Kohonen em 1982, sendo considerado relativamente simples e com a capacidade de organizar dimensionalmente dados complexos em grupos (clusters), de acordo com suas relações.

detalhes serão vistos na seção 1.3.1.

A linguagem DHJOG contempla aspectos táticos de jogos de tabuleiro envolvendo três dimensões: material, espacial e temporal. A dimensão material calcula a diferença entre o somatório dos valores associados a cada tipo de peça de cada jogador presente no tabuleiro. Uma diferença positiva indica vantagem, negativa desvantagem e zero indica equilíbrio material. A dimensão espacial envolve os conceitos de regiões de tabuleiro, para especificar conceitos posicionais (e.g. domínio de centro). E por último, a dimensão temporal permite definir as transições de etapa ou fases de um jogo (e.g. início, meio e fim), possibilitando que avaliações heurísticas diferentes sejam usadas para cada etapa.

A ferramenta HeuChess possibilita estabelecer, no momento, até quatro níveis de complexidade no uso da linguagem DHJOG, de um total de seis níveis projetados:

1. permite apenas definir um valor inicial para cada tipo de peça (dimensão material);
2. permite definir regiões no tabuleiro e criar heurísticas para modificar o valor das peças (dimensão espacial e material);
3. permite criar heurísticas gerais para valorar o tabuleiro como um todo (dimensão espacial e material);
4. permite definir etapas de jogo (dimensão temporal) e as regras para transição entre elas. A dimensão temporal está relacionada ao desenvolvimento rápido das peças;
5. possibilita alterar a expressão de cálculo heurístico de cada etapa.
6. permite alterar e criar novas funções.

E por fim, Bueno [11] apresenta uma ferramenta de apoio ao aperfeiçoamento de heurísticas táticas, através de análises estatísticas gráficas e modelagem matemática dos resultados. Permitindo aos autores da linguagem DHJOG testarem e avaliarem suas heurísticas em várias posições de jogo, previamente classificadas por enxadristas ou *engines* de xadrez reconhecidas pela comunidade enxadrística.

Todos estes trabalhos são parte de uma grande área da ciência da computação que se denomina Inteligência Artificial (IA) .

A IA pode ser definida em duas dimensões (1,2), cada uma com duas camadas (a,b): (1a) pensar como um ser humano, (1b) agir como um ser humano, (2a) pensar racionalmente e (2b) agir racionalmente. A dimensão 1 envolve a ciência empírica sobre o pensamento e comportamento humanos, e a dimensão 2 envolve as ciências da matemática e da engenharia. Contudo, outras áreas contribuem para os avanços da IA: a Filosofia, a Economia, a Neurologia, a Linguística, a Engenharia da Computação, entre outras [58].

A IA foi dividida em duas grandes categorias ou paradigmas: (1) IA simbólica e (2) IA conexionista [68]:

- A IA simbólica tem como base a chamada *folk psychology* (o conjunto de estruturas habitualmente utilizado pelo senso comum para descrever, explicar, prever e avaliar as atitudes e o comportamento das pessoas) formalizando-a como uma "linguagem do pensamento" que segue as regras de operação definidas pelas "leis do pensamento" de Boole. O conjunto de estruturas da *folk psychology* é composto de três elementos: objetivos, crenças e conceitos e são traduzidos na forma de um sistema de símbolos físicos que pode ser manipulado por uma máquina de

processamento simbólico. Lisp e Prolog são exemplos de linguagens desenvolvidas neste paradigma que permitem a criação de sistemas inteligentes.

- A IA conexionista seria o resultado do aprofundamento do sistema simbólico humano até chegar a um nível similar ao dos neurônios, também conhecido como nível sub-simbólico. As redes neurais são exemplos de sistemas deste paradigma.

Este trabalho está inserido no paradigma da IA simbólica.

1.2 Caracterização do problema

O trabalho da presente tese é uma continuidade dos já realizados, a fim de tratar o problema do conhecimento e percepção estratégicos em jogos adversaristas. O domínio de aplicação é o mesmo dos demais trabalhos (jogo de xadrez), conforme definição a seguir.

1.2.1 Definição

O jogo de xadrez tem sido usado no ensino por se tratar de um jogo que permite treinar e adquirir habilidades como: raciocínio estruturado (lógica), percepção espacial (conceitos visuais), tomada de decisões (qual peça mover), memorização (situações de jogo), entre outros ([70], [18], [13], [15]).

No que diz respeito à tomada de decisões, cabe ao jogador escolher qual o melhor movimento dado em um determinado contexto de jogo. Em média existem 35 possibilidades de movimentos para cada situação. Esta decisão deve ser tomada de acordo com uma tática, aqui entendida como heurística de jogo. Um exemplo seria apenas considerar a vantagem material. Neste caso a decisão seria buscar por movimentos de captura de peças do adversário e proteção das próprias. Apesar de ser uma tática válida não significa que o resultado será uma vitória.

Do ponto de vista estratégico, entretanto, o tema é controverso e de difícil definição. Mas de maneira geral entende-se que se trata de uma visão de futuro baseado em um plano mental. Segundo o site especializado <https://www.chessprogramming.org/Strategy> (acesso em 05/12/2018), diz respeito ao jogo posicional², a definição de um alvo e planos de longo prazo, levando em consideração diversos elementos táticos/posicionais como estrutura de peões, segurança do rei, roque por fazer, etc.

Como será visto na revisão de literatura não se trata de um problema e nem de uma solução trivial, pois, refere-se a um domínio mal definido/estruturado. Nem mesmo na literatura especializada existe uma clara definição do que seja considerado um elemento heurístico puramente estratégico e do que seja considerado puramente tático. Poucos trabalhos correlatos foram encontrados (seção 2.5).

O que se sabe, através de reuniões com enxadristas experientes, é que a estratégia está ligada ao lado abstrato (intuição, intenções, percepções). A tática, por sua vez, com o lado concreto do jogo (elementos visíveis), assim todo movimento é uma ação tática, mas que devidamente sequenciados produzem resultados no nível estratégico.

Assim sendo, no contexto deste trabalho, assume-se que uma estratégia é o delineamento de um plano com objetivo de conduzir a partida para uma posição de jogo

²Jogo dominado por manobras de longo prazo vantajosas, ao invés de ataques e ameaças de curto prazo, exigindo a habilidade de julgamento e não apenas cálculos das variações de um movimento.

que se encaixe em um padrão preestabelecido. Um plano estratégico pode ser mono-escalonado ou multi-escalonado [34] *apud* [15, p.79]. No mono-escalonado apenas um padrão de futuro é formalizado. No multi-escalonado existe uma sequência de padrões que se pretende atingir. Este objetivo normalmente está ligado a ideia de: (a) modificar o balanço das forças, (b) obter o controle da posição ou (c) tentar induzir o jogo para um tipo de final.

Outro exemplo seria induzir situações que requeiram a execução de um ataque duplo (quando uma única peça ameaça duas ou mais peças simultaneamente), o qual pode ser executado por diferentes peças (bispo, torre, cavalo, dama e peão - o ataque duplo deste último também é chamado de garfo). No caso um dos componentes usados na estratégia poderia ser optar por posições onde existam ataques duplos contra peças como dama e rei, a fim de trazer ganho material, modificando o balanço de forças do jogo.

Os estudos realizados por DeGroot [18], mostram que o pensamento estratégico advém do acúmulo de conhecimento de padrões úteis e de características chave em uma posição. Simon [65], confirma estes estudos e, vai além, percebendo que estes padrões são armazenados em blocos interconectados por uma complexa rede de associações. Assim um padrão também pode ser visto como uma conjunção de blocos que cooperam entre si.

Estes padrões são acessados na memória de longo prazo de um grande mestre durante o desenrolar de uma partida. Para efeitos estratégicos a busca é por padrões que sejam consequentes de outros, de onde surge a questão fundamental deste trabalho: é possível criar um modelo de formalismo e representação do conhecimento pelo qual seja possível identificar um padrão e usá-lo como parâmetro de busca?

Linhares [40] realizou um trabalho sobre cenários estratégicos e o papel de pedaços relevantes de informação (que é conhecido com o termo *chunks* conforme definição na página 53, apresentando críticas ao trabalho de Simon e do conceito de memória POS (Piece-on-Square). Apresenta também evidências que os grandes mestres conseguem, mesmo em posições bem diferentes uma das outras, perceber similaridades estratégicas entre elas. O trabalho propõe que isto está no nível de visão estratégica como resultado da percepção dos papéis abstratos desempenhado pelas peças.

No xadrez, bem como em outros jogos de tabuleiro, existem muitos padrões conhecidos, dentre estes os padrões de xeque-mate são bastante relevantes para este estudo, pois podem ocorrer em posições muito diferentes uma das outras, desde os movimentos iniciais (abaixo de 15 lances) até nas fases finais da partida. Linhares [40] utilizou um destes padrões em seus experimentos com enxadristas, o qual é conhecido como Mate Philidor.

A figura 1.1 apresenta duas variações do Mate Philidor com a sequência de movimentos necessária para execução do padrão. Muito embora as posições iniciais de cada sequência sejam diferentes, humanos enxadristas experientes percebem as similaridades entre elas. Entretanto, computadores não percebem tais semelhanças, pois se baseiam essencialmente em conhecimento tático [28].

Nas duas variações a sequência de movimentos obedece um mesmo padrão. Na posição inicial percebe-se que uma peça branca consegue atacar o Rei preto, que está afogado, através de um movimento. Após este movimento, o oponente, forçadamente com a sua torre, deverá capturar a peça branca que ameaça seu Rei, deixando-o completamente afogado (sem opções de jogo). Em seguida, o cavalo branco se moverá para executar o xeque-mate, o que pode ser verificado nas posições finais.

Considerando o exposto até aqui, um padrão passa a ser definido como: dada uma posição de tabuleiro, um subconjunto de peças que exercem relações específicas entre

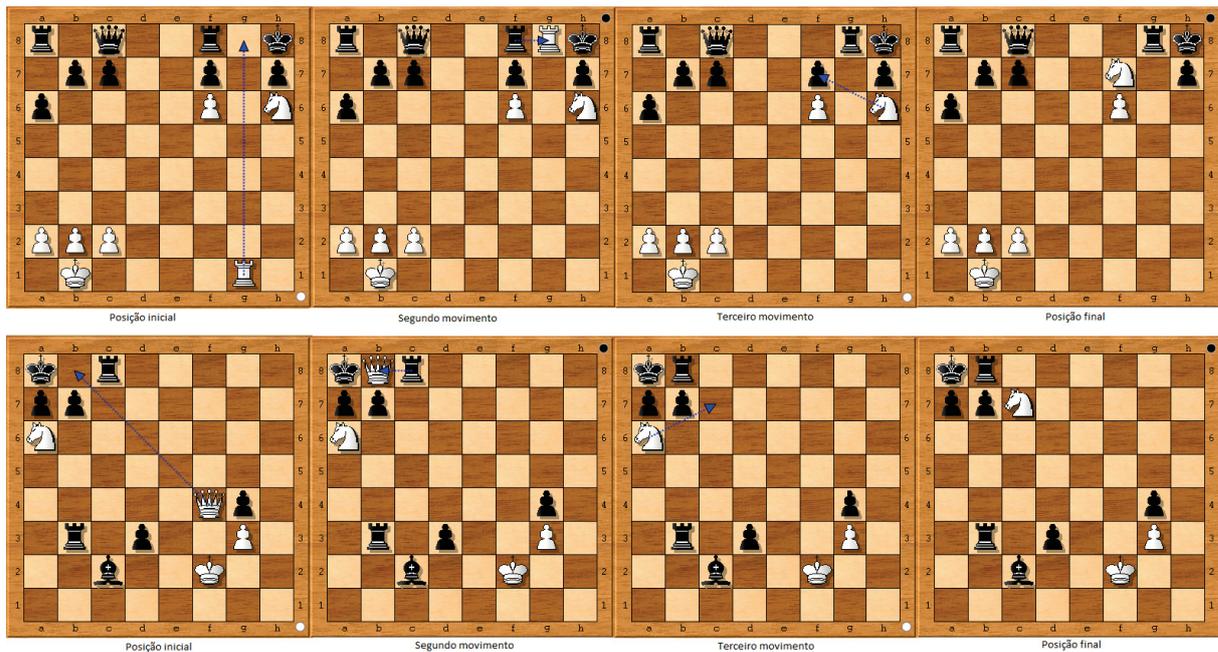


Figura 1.1: Similaridade em nível de estratégia - Fonte: adaptado de [40]

si, cujos papéis desempenhados são abstratos, no sentido de que podem ser exercidos por peças diferentes, em posições bem diferentes, mas cujos movimentos e resultados são similares.

Padrões como o Philidor, além de um cenário específico, dependem de outros fatores condicionais e ou testes conceituais que precisam estar presentes no cenário inicial. E em alguns casos (Philidor), por se tratarem de padrões temporais, condições específicas também precisam ser avaliadas no estado terminal do padrão.

Doravante, considerando este conceito, um padrão passa a ser denominado Padrão de Alto Nível, cujo acrônimo, será transliterado para a língua inglesa como HLP (High Level Pattern), para efeito de publicações futuras. Tanto o termo "HLP" quanto o termo "padrão" serão considerados como sinônimos neste texto. Padrões que atendem estas características são complexos, exigem um esforço de raciocínio para identificar como eles são, quando surgem, que condições são necessárias para que o padrão esteja presente, quais as táticas envolvidas, como ele evolui no tempo e qual o seu estado final. Somente tendo este entendimento é possível representá-los de forma adequada.

Apesar da definição adotada, existem padrões no nível de visão estratégica que se identificam apenas pela prova de conceitos, sem necessariamente especificar movimentos táticos e verificação de estados finais.

O ponto central deste conceito é que uma estratégia surge a partir da identificação do padrão presente na posição e na conexão mental com um estado futuro de vantagem/desvantagem. No caso do padrão Philidor, ao perceber o cenário inicial, o enxadrista já percebe a vitória absoluta. Isto é tão verdadeiro, que nos jogos usados nos experimentos deste trabalho, alguns oponentes ao perceberem que o padrão estava se configurando, desistiam antes mesmo de chegarem à posição inicial.

Por esta razão o conceito de estratégia aqui defendido fica restrito ao contexto de uma partida e é dependente do reconhecimento de padrões, e difere da noção geral de estratégia, como no caso de uma guerra ou em gestão empresarial, onde existem muitos elementos internos (baixa produtividade dos colaboradores, demissões, greves, etc) e

externos (desastres naturais, leis com alta de impostos, etc.) a serem considerados. No Xadrez o único fator relevante a considerar são os movimentos do oponente que possam frustrar a execução de seu plano.

Uma estratégia é usada para dar orientação aos movimentos e diferencia-se de tática, no sentido de que, esta diz respeito aos diversos temas, motivos e justificativas de movimento que podem existir numa posição, tais como: aplicar um garfo, um ataque-duplo, ataque a descoberto, raio-x, cravada, entre outros. A estratégia coordena as ações e a ordem de todos estes elementos.

Jogar de forma estratégica difere do jogo tático, onde o objetivo é o ganho material. Também difere, mas em menor grau, do jogo puramente posicional, onde o objetivo é ganhar espaço, pela avaliação de diversos critérios como mobilidade, controle de centro, segurança do Rei, estrutura de peões, entre outros. Significa que os movimentos serão seletivos, buscando-se os que melhor se ajustam ao plano [81]. Um exemplo de plano poderia ser coordenar um ataque para dar xeque-mate com dama e cavalo, em outras palavras, identificar cenários que propiciem tais tipos de jogada.

Conforme o que já foi colocado anteriormente, estratégia é a parte abstrata (o plano, a intenção, a percepção), acrescentamos a isto, o elemento de coordenação das ações e a execução dos lances (tática) é a parte concreta. Assim todo o movimento em si mesmo é tático, mas para efeitos deste trabalho, será considerado que um movimento é dito estratégico, quando ele está atrelado a um conjunto coordenado de ações que é estabelecido a partir da visualização/percepção antecipada de um cenário (padrão).

1.2.2 Problemas de busca e algoritmos de busca heurística

Dentro da IA existe um ramo que trata de problemas complexos teoricamente e que não são resolvidos com as técnicas tradicionais de programação. Este ramo é denominado de Problemas de Busca e sua complexidade depende do tamanho do espaço onde a busca deve ser realizada. Os problemas de busca residem em satisfazer um objetivo ou encontrar um estado-solução dentro de espaço de estados e devem ter um objetivo bem definido, um estado inicial e um conjunto de regras que estabeleçam as mudanças de um estado para outro [58].

Problemas de busca são tratados através de algoritmos que dado um estado inicial dentro do espaço de busca, um estado-solução ou objetivo bem definidos, consideram um conjunto de ações que alteram o estado inicial levando a um novo estado, produzindo o que é conhecido como árvore de busca na forma de um grafo. O estado inicial é chamado de nó raiz. Os estados gerados a partir do nó raiz são chamados de nós-filhos. Para cada nó-filho um novo conjunto de ações é gerado de acordo com as regras do problema. A quantidade de nós-filhos que cada estado pode gerar é denominado de fator de ramificação [58].

A forma como a busca será executada em uma árvore é denominada de estratégia de busca e pode ser de dois tipos: busca em largura ou busca em profundidade. A busca em largura expande todos os nós-filhos de um estado e avalia cada estado antes de gerar a próxima sequência de de nós-filhos. A busca em profundidade expande o primeiro nó-filho do estado inicial e o faz sucessivamente até encontrar um estado que não possua filhos, ou ter encontrado o estado-solução ou ainda ter alcançado um limite de profundidade por questões de tempo de processamento [58] ou de memória.

Sob o aspecto da inteligência para tratar a busca os algoritmos se dividem em duas categorias: (1) busca às cegas, ou seja, sem considerar nenhum conhecimento prévio sobre o problema ou (2) busca usando algum conhecimento prévio sobre o problema que

seja útil na avaliação de um estado e que indique quão bom ele é para encontrar um estado-solução. A categoria 2 é também conhecida como algoritmos de busca heurística. O algoritmo conhecido como MINIMAX³, por exemplo, se encontra nesta categoria de algoritmos e é o mais utilizado em jogos como o Xadrez.

O conhecimento utilizado no MINIMAX atua no nível tático e posicional. Dada uma posição, primeiramente geram-se os movimentos legais, os quais são ordenados por critérios táticos como: captura de peças ou número de ameaças resultantes do movimento. Ativa-se o algoritmo recursivamente passando a nova posição encontrada, repetindo o processo até a profundidade de busca máxima ser alcançada. As posições finais são avaliadas pela Função de Avaliação Estática (FAE) que realiza o cálculo material e posicional e retorna um valor indicando quão promissora a posição é para chegar ao xeque-mate que é o objetivo final do jogo.

O MINIMAX produz uma árvore de busca tipo E-OU [58] e no caso do xadrez, o fator de ramificação é de 35 nós-filhos em média para cada estado [18]. Cada estado é uma posição de jogo resultante de um movimento. Assim, a cada nível de profundidade ocorre um aumento gigantesco de estados possíveis, o que acaba inviabilizando a busca em toda a árvore, mesmo para máquinas muito rápidas e com muita memória. Uma árvore de decisão com altura 10, apenas 5 lances completos, teria o equivalente a 35^{10} nós para serem avaliados. Partidas de xadrez podem atingir facilmente 40 lances (80 níveis). A maior partida, oficialmente registrada no *Guinness Book* [7], durou 20 horas e 15 minutos, chegando a 269 lances completos.

Este algoritmo trabalha bem em situações nas quais táticas de curto prazo são prevalentes e sua FAE consegue identificar as consequências dos movimentos, dentro de seu horizonte de busca. Entretanto é fato que existem posições que exigem visão estratégica para aceitá-las ou rejeitá-las, em tais situações as limitações do algoritmo surgem [59].

É claro que, se a busca for aprofundada, através do cálculo tático MINIMAX é possível encontrar movimentos que podem se entendidos como estratégicos. Entretanto, isto depende dos fatores heurísticos que são avaliados pela FAE. Van Reek [81] apresenta um estudo sobre jogadores automáticos que conseguiram localizar tais movimentos em alguns casos.

Existem vários exemplos em partidas que ajudam a compreender a diferença entre um movimento puramente tático de um com viés estratégico. Um exemplo clássico ocorreu na partida entre Nigel Short x Jan Timman (1991). Conforme Marler [43] argumenta, após o lance 31, Short estabeleceu como estratégia provocar um mate usando a dama e o rei, para isto ele utilizou um conceito abstrato conhecido como "Caminhada do Rei", com o objetivo de levá-lo até a casa h6, veja a Figura 1.2, assim o movimento escolhido foi Rg3.

A fim de verificar o que foi dito por Herbst [28] sobre as *engines* atuais se basearem apenas em conhecimento tático, o seguinte experimento foi executado neste trabalho: com o auxílio do programa Arena 3.5 e a *engine* Stockfish 6, considerada uma das melhores de código aberto que existe, segundo a própria equipe que a desenvolveu (<https://stockfishchess.org/> - acesso em 02/12/2018). Stockfish utiliza o algoritmo MINIMAX com podas alfa-beta como base, mas também utiliza aprimoramentos como busca com profundidade iterativa, *Quiescence Search*⁴, *Principal Variation (PV)*

³O algoritmo MINIMAX foi desenvolvido pelo matemático John Von Neumann em 1928 [83].

⁴*Quiescence search* é uma técnica usada no MINIMAX para evitar cair em armadilhas. Geralmente usada quando há intensa atividade de trocas de peças ou quando a FAE avalia muito bem uma posição. Nestes casos a busca é aprofundada para ver se a vantagem se mantém ou não existam mais trocas sucessivas.

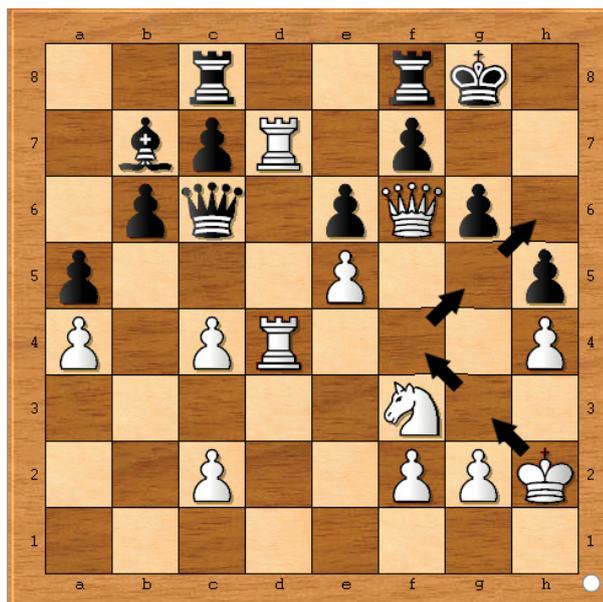


Figura 1.2: Movimento tático x estratégico - brancas jogam - Fonte: adaptado de [43]

*search*⁵, entre outras, para aumentar o desempenho e a qualidade da busca.

Pelo fato do xeque-mate com Rei e Dama estar 14 movimentos a frente (*plies*⁶), o algoritmo foi configurado para uma profundidade de 16. Ao solicitar a análise e execução do movimento, a máquina Stockfish sugeriu c3 com avaliação +0,91. Devido a *quiescence search* em alguns nós a profundidade atingida foi de 24, contudo sem encontrar uma posição de vitória para as brancas. Ao reiniciar a posição e solicitar uma reanálise o algoritmo sugeriu Df4 como alternativa.

Em seguida, após reiniciada a *engine* Stockfish, para zerar a *cache* e a tabela *hash*, um novo teste foi executado com profundidade de 32. O movimento escolhido neste caso foi Rg1 com +1,02. Fica evidente que para o cálculo tático desta *engine* a variação Rg3 não é considerada.

Para fazer com que a Stockfish escolhesse o movimento Rg3 foi necessário ativar a avaliação Multi-PV com largura de ramificação 10, que por padrão estava desativada. Ou seja, o teste foi configurado para que o algoritmo avaliasse as sub-árvores dos 10 melhores movimentos ordenados no primeiro nível. Nestas condições, ao atingir a profundidade 31 (mais de 15 movimentos completos), finalmente a variação Rg3 foi encontrada com avaliação positiva de +6,87, ainda assim sem encontrar o mate na busca por força bruta. Depois de encontrado este movimento, o algoritmo reproduziu os movimentos realizados por Short.

O importante a destacar, no exemplo da Figura 1.2, é que Short percebeu estrategicamente a vitória pela percepção do padrão conceitual citado acima. Isto reforça o conceito de estratégia que está sendo proposto aqui.

As heurísticas de ordenação, PV, podas alfa-beta e o conhecimento tático de sua FAE tornam o MINIMAX altamente eficaz para reduzir o escopo de busca, em contrapartida, o tornam inapropriado para encontrar certas posições onde existe vitória. A prova

⁵método de busca que depende de um bom método de ordenação que classifique quais movimentos são melhores. O melhor movimento é denominado de PV e é escolhido para aprofundar a busca MINIMAX.

⁶Ply é o termo usado pela comunidade de programadores que designa o movimento de uma peça durante uma partida, diferentemente da ideia de lance que contabiliza um movimento das brancas e outro das pretas.

deste argumento pode ser visto na tabela 1.1, que mostra os valores retornados pela FAE da máquina Stockfish 6.0 para as duas posições do trabalho de Linhares [40] referentes ao padrão Philidor (posições iniciais da figura 1.1, respectivamente).

Nos dois casos o valor resultante da avaliação estática é negativo, o que significa

Tabela 1.1: Avaliação estática - Posições Philidor

Posição FEN	Valor FAE
r1q2r1k/1pp2p1p/p4P1N/8/8/PPP5/1K4R1 w - - 0 1	-11,90
k1r5/pp6/N7/8/5Qp1/1r1p2P1/2b2K2/8 w - - 0 1	-3,65

vitória para o oponente, o que não é verdadeiro, pois as brancas jogam e há uma sequência forçada que leva ao xeque-mate, desde que aplicada a tática correta. Usando os mecanismos de ordenação PV e podas alfa-beta estas posições seriam evitadas ou até mesmo não encontradas na busca. No Capítulo 5 mais dados serão apresentados a fim de reforçar esta afirmação.

1.3 Representação de conhecimento

Dentro da IA, a representação de conhecimento (RC) é um dos pontos centrais para a modelagem e solução de qualquer problema. Mas, exatamente o que significa representar conhecimento? Quais as maneiras de fazê-lo?

Para Davis [17], isto pode ser entendido pela distinção dos papéis desempenhados e que são cruciais para a solução de um problema. Pelo menos cinco papéis são descritos em seu artigo:

1. Uma RC deve atuar como um substituto para o elemento em si mesmo.
2. Atua como um conjunto de afirmações ontológicas para responder uma questão.
3. É uma teoria fragmentada do raciocínio inteligente em termos de: (1) concepção fundamental da representação do raciocínio inteligente, (2) inferências que sanciona e (3) inferências que recomenda.
4. É um ambiente computacional de simulação onde o pensamento é realizado. É um meio para a computação eficiente.
5. É um meio de expressão humana, ou seja, uma linguagem onde são feitas asserções sobre o mundo em questão.

Para a presente tese os papéis 1, 2, e 5 das RCs se destacam:

O item 1 é importante pois um dos objetivos (veja a seção 1.6.2) é a criação de uma linguagem para descrever HLPs de tal forma que olhando para a representação simbólica se tenha uma compreensão de como o padrão é no mundo real. O item 2 se destaca, pois para compreender os elementos importantes de um HLP foi definida uma ontologia, com base na qual foram criados os elementos da linguagem de representação de HLPs. O item 5 é importante, pois a linguagem também foi definida para humanos explicitarem seu conhecimento e melhorar sua percepção sobre quais elementos são importantes no HLP objeto de sua representação.

Sob o aspecto da substituição, um dos pontos cruciais é saber quão próximo a RC

está da coisa real. Que atributos são abstraídos e tornado explícitos e quais não. Segundo Davis [17] é impossível, de forma geral, alcançar um modelo fiel por completo, tanto na prática quanto em princípio. Em princípio porque a melhor representação de um objeto é o próprio objeto. Assim qualquer representação de algo é, em princípio, imprecisa, podendo, conseqüentemente ser uma fonte de erros.

No aspecto ontológico, adotar um modelo de representação significa fazer um conjunto de afirmações ontológicas. Isto funciona como um filtro que especifica o que será visto com acuidade, devido a sua relevância, com a contrapartida de desfocar e ignorar outras partes. Porém, aqui reside uma das mais importantes contribuições de uma RC, que é a especificação de ontologias sobre um domínio de conhecimento [17].

Ontologias, podem ser descritas em uma variedade de linguagens e notações. A forma em si, como é feita, não é essencial, mas sim o seu conteúdo. O conteúdo é o que fornece subsídios para simular o pensamento humano.

1.3.1 Formalização do conhecimento heurístico

Pensar de forma estratégica exige habilidades de análise das condições presentes e da escolha correta de um caminho a ser seguido, entre vários possíveis, visando atingir uma determinada condição futura de maior vantagem.

Tubau [80] demonstra que através de uma representação explícita é possível inferir corretamente o caminho a ser seguido, maximizando o resultado da função de avaliação. O estudo de Tubau utiliza o problema conhecido como *Monty Hall* no campo da análise de probabilidades para melhorar o raciocínio probabilístico dos participantes num processo de tomada de decisão. Conclui que a habilidade de imaginar e representar de forma completa e explícita as diferentes possibilidades de escolhas afeta fortemente a acurácia do raciocínio probabilístico.

Assim, as pesquisas que vem sendo desenvolvidas pelo Departamento de Informática da UFPR consistem na criação de ferramentas de *software* e modelos conceituais que ajudem no processo de explicitação de conhecimentos heurísticos em jogos adversaristas por pessoas leigas em programação de computadores.

Feitosa *et. al.* [22] demonstra, de forma empírica, que a explicitação formal do conhecimento heurístico melhora a compreensão, por parte de aprendizes, dos elementos táticos que são importantes para a avaliação de uma jogada. Assim, é razoável pensar que, o formalismo de um HLP, no nível da visão estratégica, também é positivo para o entendimento de como: ele ocorre, quais são suas partes, qual sua relevância durante uma partida, quais são seus aspectos espaciais, quais seus aspectos temporais e seu estado final quando for o caso.

A seguir serão apresentados quatro exemplos da DHJOG na representação explícita de conceitos para os níveis 1, 2, 3 e 4 de expressividade implementados (seção 1.1). Todos os componentes heurísticos ficam encapsulados dentro de uma grande estrutura denominada de CONJUNTO HEURISTICO.

A Listagem 1.1 descreve como fica definida a valoração inicial de peças. Neste caso só existe uma etapa definida para todo o conjunto heurístico, que recebe o nome de etapa PADRAO (linha 5). A partir da linha 7 são definidos o valor das peças e na linha 12 é definido qual será o valor heurístico do tabuleiro para a etapa PADRAO, que neste caso é o somatório dos valores das peças presentes no tabuleiro do jogador que está avaliando a posição (EU) menos o somatório dos valores das peças presentes no tabuleiro do jogador oponente.

```

1  CONJUNTO_HEURISTICO "EXEMPLO_NIVEL01"
2  ANOTACAO "EXPLICACAO INICIAL" EXPLICACAO
3  "Exemplo de heurística de nível 01"
4  FIM ANOTACAO
5  ETAPA_ATUAL <- "PADRAO"
6  ETAPA "PADRAO"
7  PEAO.VALOR <- 1
8  TORRE.VALOR <- 5
9  CAVALO.VALOR <- 7
10 BISPO.VALOR <- 5
11 DAMA.VALOR <- 9
12 TABULEIRO <-
13   SOMA_PECAS({PEAO,TORRE,CAVALO,BISPO,DAMA},TODO_TABULEIRO,EU) -
14   SOMA_PECAS({PEAO,TORRE,CAVALO,BISPO,DAMA},TODO_TABULEIRO,OPONENTE)
15 FIM ETAPA
16 FIM CONJUNTO_HEURISTICO

```

Listagem 1.1: Exemplo nível 01 - valoração inicial de peças

Na listagem 1.2, além da valoração inicial de peças definidas para a etapa padrão, está definida uma região denominada CENTRO composta pelo conjunto de casas expresso na linha 16, e uma heurística para alterar o valor heurístico dos cavalos que estiverem posicionados nesta região (linhas 17 a 22).

```

1  CONJUNTO_HEURISTICO "EXEMPLO_NIVEL02"
2  ANOTACAO "EXPLICACAO INICIAL" EXPLICACAO
3  "Exemplo de heurística de nível 02"
4  FIM ANOTACAO
5  ETAPA_ATUAL <- "PADRAO"
6  ETAPA "PADRAO"
7  PEAO.VALOR <- 1
8  TORRE.VALOR <- 5
9  CAVALO.VALOR <- 7
10 BISPO.VALOR <- 5
11 DAMA.VALOR <- 9
12 TABULEIRO <-
13   SOMA_PECAS({PEAO,TORRE,CAVALO,BISPO,DAMA},TODO_TABULEIRO,EU) -
14   SOMA_PECAS({PEAO,TORRE,CAVALO,BISPO,DAMA},TODO_TABULEIRO,OPONENTE)
15
16  CASA[] "CENTRO" <- {c6,d6,e6,f6,c5,d5,e5,f5,c4,d4,e4,f4,c3,d3,e3,f3}
17  HEURISTICA VALOR_PECA "CAVALONOCENTRO"
18  SE
19   PRESENCA_PECAS({CAVALO},CENTRO,EU) IGUAL VERDADEIRO
20  ENTAO
21   PECAS_QUE_ESTAO({CAVALO},CENTRO,EU) + 2,0
22 FIM HEURISTICA
23 FIM ETAPA
24 FIM CONJUNTO_HEURISTICO

```

Listagem 1.2: Exemplo nível 02 - região e valoração de peças tabuleiro

A Listagem 1.3 mostra como escrever uma heurística que será ativada caso o rei oponente esteja em xeque e exista um peão que consiga ser inevitavelmente promovido na posição de jogo em análise, o que representa uma vantagem posicional. Para isto é necessário definir em quais casas o peão é promovido (linha 16) e quais as casas próximas (linha 17). Das linhas 18 a 25 está a definição desta heurística. Se existem um peão prestes a promover, o rei está em xeque e não existem peças do oponente na linha de promoção o valor do tabuleiro é multiplicado por 2.

```

1  CONJUNTO_HEURISTICO "EXEMPLO_NIVEL03"
2  ANOTACAO "EXPLICACAO INICIAL" EXPLICACAO
3  "Exemplo de heurística de nível 03"
4  FIM ANOTACAO
5  ETAPA_ATUAL <- "PADRAO"
6  ETAPA "PADRAO"
7  PEAO.VALOR <- 1
8  TORRE.VALOR <- 5
9  CAVALO.VALOR <- 7
10 BISPO.VALOR <- 5
11 DAMA.VALOR <- 9
12 TABULEIRO <-
13   SOMA_PECAS({PEAO,TORRE,CAVALO,BISPO,DAMA},TODO_TABULEIRO,EU) -
14   SOMA_PECAS({PEAO,TORRE,CAVALO,BISPO,DAMA},TODO_TABULEIRO,OPONENTE)
15
16  CASA[] "PROMOCAO" <- {a8,b8,c8,d8,e8,f8,g8,h8}
17  CASA[] "PROXIMO" <- {a7,b7,c7,d7,e7,f7,g7,h7}
18  HEURISTICA VALOR_TABULEIRO "PROMOVERPEAO"
19  SE
20   PRESENCA_PECAS({PEAO},PROXIMO,EU) IGUAL VERDADEIRO E
21   ESTA_AMEACADA({REI},TODO_TABULEIRO,OPONENTE) E
22   PRESENCA_PECAS({REI,CAVALO,TORRE,DAMA,BISPO},PROMOCAO,OPONENTE) IGUAL FALSO

```

```

23     ENTÃO
24     TABULEIRO <- TABULEIRO * 2
25     FIM HEURISTICA
26     FIM ETAPA
27     FIM CONJUNTO_HEURISTICO

```

Listagem 1.3: Exemplo nível 03 - valoração do tabuleiro

Finalmente, a Listagem 1.4 apresenta um exemplo de heurística de transição de etapa. Existem duas etapas definidas: ABERTURA e MEIODEJOGO. Se a quantidade de lances (jogadas) for maior ou igual a 10, ocorre a transição para a etapa MEIODEJOGO. Evidentemente esta não é uma boa heurística, porém foi colocado como exemplo de formalismo da linguagem que um humano aprendiz poderia escrever. No momento da colaboração no processo de ensino, este conceito seria então revisto a fim de propor uma heurística de melhor qualidade.

```

1  CONJUNTO_HEURISTICO "EXEMPLO_NIVEL04"
2  ANOTACAO "EXPLICACAO INICIAL" EXPLICACAO
3  "Exemplo de heurística de nível 04"
4  FIM ANOTACAO
5  ETAPA_ATUAL <- "ABERTURA"
6  ETAPA "ABERTURA"
7     PEAO.VALOR <- 1
8     TORRE.VALOR <- 5
9     CAVALO.VALOR <- 7
10    BISPO.VALOR <- 5
11    DAMA.VALOR <- 9
12    TABULEIRO <-
13    SOMA_PECAS({PEAO,TORRE,CAVALO,BISPO,DAMA},TODO_TABULEIRO,EU) -
14    SOMA_PECAS({PEAO,TORRE,CAVALO,BISPO,DAMA},TODO_TABULEIRO,OPONENTE)
15
16    HEURISTICA TRANSICAO_ETAPA "PARAMIODEJOGO"
17    SE
18    QUANTIDADE_JOGADAS() >= 10
19    ENTÃO
20    ETAPA_ATUAL <- "MEIODEJOGO"
21    FIM HEURISTICA
22    // demais heurísticas de transição de valor de peças e posição de jogo
23    FIM ETAPA
24    FIM CONJUNTO_HEURISTICO

```

Listagem 1.4: Exemplo nível 04 - transição de etapa

A ferramenta e a linguagem idealizadas por Feitosa permitem a autoria dos elementos heurísticos que serão avaliados dentro da FAE.

No Capítulo 4 serão vistos novos formalismos idealizados para cobrir aspectos importantes do ponto de vista estratégico.

1.4 Justificativa

É fato conhecido que a aquisição de perícias em qualquer nível de aprendizado depende muito da prática realizada através de exercícios propostos, cuja solução já se conhece. Adquirir habilidades estrategistas não foge à regra.

Soltis [70] afirma que a grande maioria das pessoas que leva a sério o jogo de xadrez atinge uma parede em seu *rating* ELO⁷. Alguns enfrentam esta parede com 1500, outros 1700 e ainda outros em níveis acima de 1700. Afirma também que apenas 2% dos que se dedicam acabam se tornando grandes mestres⁸.

Para Soltis uma explicação para esta parede seria:

⁷*Rating* ELO é um método criado por Arpad Elo, utilizado pela Federação Internacional de Xadrez (FIDE) para se calcular a força relativa entre enxadristas.

⁸Grande Mestre Internacional: Título outorgado pela FIDE aos enxadristas que alcançam um conjunto de padrões estabelecidos, incluindo a classificação ELO. É o mais alto título (além de Campeão Mundial) que pode ser obtido no xadrez.

"...muitos jogadores chegam onde estão pelo aprendizado de como não perder. Quando dois jogadores no nível de 1700 ELO se encontram sobre um tabuleiro, um irá tipicamente se auto-destruir. Não na abertura como um jogador de 1100 faria, mas pelo movimento 40, se não no 30. Como resultado, muitos jogadores podem alcançar 1900 simplesmente por não cometer erros estúpidos." [70, pp. 6]

A existência de poucas ferramentas voltadas para o ensino de xadrez (ver seção 2.3) é um fator motivador para a execução deste trabalho. Além disto, não foram encontradas na literatura ferramentas de autoria voltadas para análise e formalização de padrões no nível de visão estratégica.

Outro ponto importante que justifica este trabalho é que, apesar da linguagem DHJOG possuir diversos componentes heurísticos, se limita apenas a avaliação estática de uma posição. Assim, não há suporte para a especificação de padrões genéricos com dinâmica de movimentos bem como um método para a detecção numa posição de jogo. Pela linguagem DHJOG situações como, por exemplo a explicitação de ataques descobertos, raio-x, padrões de mate, entre outros, se torna impraticável pela falta de mecanismos e símbolos que deem suporte para tais representações.

Além do exposto acima, a inexistência de um formato na notação enxadrística que permita descrever um cenário genérico envolvendo ações estáticas e temporais simultaneamente, e descrever uma posição em função dos papéis desempenhados pelas peças é uma lacuna a ser preenchida.

Na notação enxadrística atual, os modelos mais utilizados são: (1) o *Portable Game Notation* (PGN) para representar os movimentos realizados numa partida. Os movimentos são representados no formato algébrico. (2) O formato *Forsyth-Edward Notation* (FEN) utilizado para descrever uma determinada posição de peças em um tabuleiro de xadrez.

E por fim, sob o aspecto dos algoritmos de busca, também justifica-se este trabalho, pois o conhecimento representando na FAE baseia-se fortemente na vantagem material, no posicionamento de peças e táticas como domínio de centro. Isto é comprovado pela avaliação resultante em posições onde existe um padrão de xeque-mate ser negativa com vantagem para o oponente ou com pouca ou nenhuma vantagem decisiva. No Capítulo 5 serão apresentados mais dados que demonstram tal argumentação.

1.5 Hipóteses de pesquisa

Este trabalho estabelece 2 hipóteses. A primeira é de que é possível representar, de forma genérica, padrões no nível de visão estratégica, e usar tais representações, durante uma busca. A segunda é de que com base no conhecimento formalizado é possível identificar se duas posições distintas são similares do ponto de vista da visão estratégica.

1.6 Objetivos

Para verificar a validade da hipótese acima, os seguintes objetivos foram definidos:

1.6.1 Objetivo geral

Definir um modelo de representação de padrões na forma de uma linguagem livre de contexto e um algoritmo de interpretação e localização do padrão dada uma posição de jogo como entrada, para que humanos reflitam sobre seus conhecimentos de padrões no nível de visão estratégica através da explícita formalização dos mesmos.

1.6.2 Objetivos específicos

- Elicitar conhecimento de especialistas humanos em reuniões para refinar os conceitos de estratégia e reconhecimento de padrões;
- Elaborar um *ambiente* para a formalização da visão estratégica;
- Criar uma nova linguagem para representação de padrões que possa ser interpretada pela máquina (hipótese 1), aqui denominada de *Pattern Language*(PATLAN) ;
- Programar o interpretador e identificador de padrões (hipótese 2);
- Construir uma interface de usuário que possibilite a autoria de padrões, permitindo ao humano refletir sobre seu conhecimento e assim poder explicitá-lo. A interface deve permitir que o humano teste seu formalismo em posições de jogo diferentes (hipótese 2);
- Testar os elementos da linguagem em situações de jogo reais existentes em bases de partidas.(hipótese 2).

1.7 Considerações finais

É importante ressaltar que não é objetivo, deste trabalho, criar uma máquina que jogue de forma estratégica por si só, mas uma metodologia para que a máquina possa sugerir a existência de um padrão, usando os formalismos da linguagem PATLAN. Além disto, criar a possibilidade de humanos representarem seus conhecimentos no nível da percepção estratégica, validá-los, avaliarem seus efeitos e refletirem sobre sua visão estratégica.

O Capítulo 2 contém uma revisão de literatura ampla, envolvendo as principais áreas de conhecimento desta pesquisa. O Capítulo 3 apresenta a metodologia adotada, bem como o conjunto de tecnologias de base. O capítulo 4 apresenta o arcabouço do *framework* e as especificações da linguagem. O Capítulo 5 contém os experimentos empíricos e resultados obtidos. E no Capítulo final as conclusões, limitações e perspectivas futuras.

2 Revisão de Literatura

O presente Capítulo apresenta os principais elementos da literatura sobre as grandes áreas e temas que envolvem esta pesquisa. Na seção 2.1 serão abordados trabalhos que se referem à Informática na Educação, onde este projeto está contextualizado e cujos resultados serão úteis para uso futuro no ensino. Na seção 2.2, uma breve revisão sobre o uso de jogos na Educação com o apoio da Informática. A seção 2.3 apresenta alguns trabalhos voltados para o ensino de xadrez com o uso de computadores. A seção 2.4 apresenta definições encontradas na literatura sobre o tema estratégia. Na seção 2.5 destina-se aos trabalhos correlatos à presente tese que tratam de máquinas de xadrez e representação de conhecimento estratégico baseado em padrões. E, por fim, a seção 2.6 apresenta artigos relativos à aprendizagem de máquina e outras abordagens diferenciadas.

2.1 Informática na Educação

Nesta seção serão revistos alguns aspectos históricos de uso da Informática na Educação, desde seus primórdios até a atualidade, em virtude que esta tese encontra-se contextualizada no PROTEX. Aqui são expostos os principais desafios, avanços e tendências previstas. Também serão vistos estudos da psicologia cognitiva que apoiam o presente trabalho como, por exemplo, o uso de múltiplas representações externas.

2.1.1 Breve história

O uso de computadores na educação é uma área de desenvolvimento que surgiu ainda na década de 1950, alguns anos após o advento dos computadores eletrônicos - MARK I (1944), seguido pelo ENIAC (*Electronic Numerical Integrator Analyzer and Computer*) (1946). As primeiras iniciativas de uso de computadores na Educação que se conhece surgiram nos EUA (Estados Unidos da América). Entretanto, nesta mesma época várias iniciativas foram encontradas em outros países como Reino Unido, Rússia, França e Bélgica [29]. Na Bélgica, inclusive, foi desenvolvido o projeto DOCEO para ensino da língua francesa, álgebra e Latim.

Em termos de literatura os primeiros documentos surgiram no início da década de 1960, a maioria na forma de relatórios sobre projetos. Segundo Hickey [29], os computadores neste tempo já eram usados para diversas atividades relacionadas à educação, tais como: controlar pagamentos, registro escolar, horários, apoio ao ensino, recuperação de informação bibliográfica, entre outras.

Nesta época começaram a surgir os primeiros sistemas de apoio ao ensino, conhecidos como instrução auxiliada por computador (IAC) (*Computer Aided Instruction*). A atividade de ensino se dá através de programas de computadores especialmente elaborados onde os alunos interagem com os equipamentos a fim de melhorar seu desempenho de

aprendizado.

O primeiro grande projeto deste tipo, descrito na literatura, teve início em 1959 na Universidade de Illinois. Donald Bitier [67] iniciou um projeto chamado PLATO. Um sistema de computador multi-terminal para uso no ensino e também em pesquisas científicas. Cada terminal tinha um teclado e uma tela sensível ao toque com recursos para mostrar caracteres, gráficos e fotos coloridas.

O sistema PLATO pode ser visto como um precursor dos atuais Sistemas Tutores Inteligentes (STI), pois contemplava algumas de suas características como:

- Um módulo de autoria, para descrever os conteúdos e as lições (exercícios) através de uma linguagem de autoria denominada TUTOR. Um módulo de interação com o estudante, onde o aluno estudava o conteúdo, executava simulações (em especial nos campos da Física e Química), e respondia um questionário ao término de cada lição.
- Um módulo do instrutor que lhe permitia entre outras coisas: a) definir a lista de alunos, b) definir um roteiro ou índice de lições a partir de um repositório central e c) acompanhar o progresso dos alunos em cada lição e respectivas pontuações nos exercícios.

Um texto publicado em 1979 [77] apresenta um vasto levantamento das iniciativas a partir dos anos 70. Pelo artigo, constata-se que houve um avanço de projetos e programas voltados para o uso da informática na educação.

Até o advento dos microcomputadores em 1975, o desenvolvimento da informática na educação foi lento devido ao custo de computadores de grande porte com sistemas operacionais de tempo compartilhado.

Aqui no Brasil, também é neste período que as primeiras iniciativas surgiram e de forma mais efetiva a partir de 1973 durante a *I Conferência Nacional de Tecnologia Aplicada ao Ensino Superior* onde foram demonstrados possíveis usos dos computadores para tal finalidade. Moraes [50] fez um resgate histórico sobre as iniciativas brasileiras.

Em 1970, Carbonell [12] introduziu um novo tipo de sistema CAI, onde pela primeira vez técnicas de IA foram usadas extensivamente. O projeto resultou num programa chamado SCHOLAR para o aprendizado de geografia, mas não exclusivamente para este fim, pois poderia ser aplicado a qualquer domínio que se desejasse estimular o aprendizado. Através de uma representação interna do conhecimento humano baseada em uma rede hierarquizada de transição entre conceitos, o programa formulava questões e recebia respostas dos alunos usando um subconjunto de palavras da língua inglesa.

A partir de então, os esforços começaram a se concentrar na integração das técnicas de IA, já conhecidas, aos sistemas CAI. Sistemas com tais características passaram a ser chamados de I-CAI (Intelligent Computer Assisted Instruction) . Tais sistemas deram origem ao que é conhecido hoje como Sistemas Tutores Inteligentes (STI), do inglês, *Intelligent Tutoring Systems (ITS)*.

Um dos principais itens da literatura sobre este vasto campo de pesquisa foi escrito por Etienne Wenger em 1987 [86]. Apesar de fazerem décadas de sua publicação, os conceitos abordados no livro continuam atuais e é um dos mais referenciados no meio acadêmico.

Em 1999, Murray [51] apresenta as principais iniciativas de pesquisa e desenvolvimento de STI. O artigo foi voltado especificamente para ferramentas de autoria para criação de sistemas tutores inteligentes.

Na sequência, serão vistos os avanços e desafios que tem sido pesquisados mais recentemente.

2.1.2 Avanços e desafios

Desde os anos 2000 até o presente outras abordagens surgiram ou ganharam maiores investimentos na tentativa de melhorar os modelos de ensino-aprendizagem dos STI. Algumas destas são [14]:

- Ensino colaborativo com suporte computacional (CSCL): pesquisa estimulada pelo potencial que a Internet tem de conectar pessoas. Geralmente combinada com o e-Learning (aprendizado à distância em boa parte) [76]. Entretanto poucas iniciativas existem que investigam os benefícios da competição junto com a colaboração, em um ambiente supervisionado através de simulações computadorizadas.
- Sistemas afetivos: o objetivo destes sistemas é agir levando em conta não apenas o aprendizado de um estudante em si mas também suas emoções. Um exemplo destas iniciativas pode ser visto em [88]. O autor defende a ideia que um STI para ser verdadeiramente afetivo, deve saber responder adequadamente as reações emocionais do aprendiz durante o aprendizado. As principais técnicas de reconhecimento de emoções envolvem colocação de sensores no corpo, reconhecimento de expressões faciais e identificação de emoções no texto.
- Agentes ensináveis: Em um STI existe um robô que tenta ensinar aprendizes humanos. Neste caso são robôs que atuam como "alunos virtuais" onde os aprendizes são estimulados a ensiná-los na resolução de algum problema específico, ou executar algum procedimento através de uma linguagem formal ou por diálogos em linguagem natural. Acredita-se que desta forma o aluno sente-se mais motivado e acaba aprendendo ao ensinar. De fato, o estudo apresentado em [38] demonstra ganhos no aprendizado dos alunos que fizeram parte do experimento.
- Aprendizado através de jogos: vários trabalhos surgiram na última década voltados para a análise, modelagem e mapeamento do aprendizado através de jogos educacionais. Um exemplo de estudo desta categoria pode ser visto em [55], onde foi demonstrado que o uso de jogos educacionais aumenta a motivação, o interesse e por consequência o aprendizado. No caso citado, o jogo é voltado para a transmissão de conceitos sobre memória computacional.
- Domínios mal definidos: anteriormente chamados de problemas mal estruturados, apresentam conceitos que podem mudar ao longo do tempo (dependência de ambiente externo) ou exigem interpretações subjetivas que tornam difícil estabelecer um método sistemático de validação das soluções encontradas. Como exemplos de tais domínios temos: legislação, arquitetura, composições musicais entre outros [41].

Além destes pontos, desenvolver sistemas de aprendizado adaptativos, independentes de domínio, multi-plataforma, com componentes reusáveis, etc, diante de uma computação cada vez mais ubíqua, são desafios relevantes. Uma iniciativa importante neste sentido é o projeto *Generalized Intelligent Framework for Tutoring* (GIFT) ([72]).

A presente tese de doutorado está inserida em duas destas abordagens de pesquisa:

1. No aprendizado através de jogos, onde a aquisição de habilidades do próprio jogo é o alvo principal da pesquisa.
2. No aprendizado dos domínios mal estruturados [64], conforme definição a seguir.

Lynch *et. al.* [41] afirma que existem muitas definições na literatura sobre domínios mal estruturados, entretanto adotam-se na presente tese os conceitos e características definidas por Simon [64], por este apresentar uma correlação deste tipo de domínio com o jogo de xadrez. Defende que jogar xadrez é um problema bem estruturado quando se trata da resolução de um único movimento (tático) mas é mal estruturado quando se trata de um jogo inteiro, ou seja, no longo prazo (estratégico). Mesmo que se encontre o movimento que maximiza o resultado para o estado atual, ainda assim, não há garantias que seja este o melhor lance possível.

Simon [64] argumenta que um problema bem estruturado tem algumas ou todas as seis condições/características abaixo:

1. Existe um critério definido e sistemático para comprovar uma solução e este pode ser feito de forma automática.
2. Existe pelo menos um espaço problema no qual pode ser representado o estado problema inicial, o estado objetivo, e todos os outros estados que podem ser alcançados, ou considerados, ao longo da tentativa de solucionar um problema.
3. Mudanças de estado atingíveis (movimentos legais) podem ser representadas em um espaço problema como transições de determinados estados para estados diretamente atingíveis a partir deles.
4. Qualquer conhecimento que o solucionador de problema pode adquirir sobre o problema pode ser representado em um ou mais espaços problema.
5. Se o problema atual consiste em atuar (agir) sobre o mundo exterior, então a definição de mudanças de estado e dos seus efeitos sobre o estado, ao se aplicar um operador qualquer, reflete com completa acurácia em um ou mais espaços (conjuntos) de problemas (leis da natureza) que governam o mundo exterior.
6. Todas estas condições mantêm um forte senso que os processos básicos postulados requerem somente volumes praticáveis de computação, e a informação postulada é efetivamente disponível para os processos - isto é, disponível com a ajuda somente de volumes praticáveis de busca.

Simon [64] também argumenta que tanto a prova de teoremas como o jogo de xadrez são problemas mal estruturados/definidos, ainda que do ponto de vista de um único movimento seja considerado como um problema bem estruturado/definido. Das seis características descritas destacam-se duas, no contexto da presente tese:

- Característica 1: no caso do xadrez isto se torna impraticável devido ao tamanho do espaço de busca. Mesmo que o conceito de "melhor movimento" seja claro, na prática ele é substituído por uma FAE que deve ser maximizada dentro de um subespaço do espaço de busca.

- Característica 5: a correspondência entre o mundo interno do pensamento e o mundo exterior de ações sobre o tabuleiro. O *feedback* de lances esperados e inesperados do oponente e as situações de tabuleiros advindas exigem recálculos e avaliações.

Simon [64] argumenta também que usar um método de planejamento pode ser visto como uma forma de tratar problemas mal estruturados.

Outra percepção importante vista no trabalho de Simon, é de que os limites (bordas) entre problemas bem estruturados e os mal estruturados são vagos e fluídos. Do ponto de vista de jogos adversaristas, tal qual o xadrez, o que seria tático em um momento poderia ser visto como estratégico em outro. Definir melhor estas bordas é uma tarefa complexa e não trivial.

2.1.3 Múltiplas Representações Externas

Segundo Davis [17], não existem formas perfeitas para representar o conhecimento, por isso autores como Ainsworth [3] defendem o uso de múltiplas formas de representar o conhecimento externamente (MRE), de forma que se tornem complementares e diminua assim o esforço cognitivo do aprendiz. Ainsworth [3], apresenta diversos estudos que evidenciam os benefícios educacionais do uso de MRE.

Ainsworth [2] descreve o *framework* DEfT, que é um conjunto de prescrições e recomendações sobre o desenvolvimento de ambientes que utilizam MRE para estimular o aprendizado de uma tarefa ou conhecimento. Ainsworth sugere uma taxonomia para MRE, dividindo em três funções (grupos) chaves de representações: (1) papéis complementares, (2) restrição de ambiguidades e (3) construção do conhecimento. Em papéis complementares uma das aplicações de MRE é na apresentação de conceitos ou informações complementares, auxiliando aprendizes a compreender todo um contexto de conteúdos.

Ainsworth defende que em ambientes de aprendizado que utilizam MRE, um conjunto de itens de projeto devem ser considerados, e são: (a) o número de representações usadas, (b) a maneira como a informação é distribuída sobre as representações, (c) a forma do sistema representacional, (d) a sequência da representação e (e) suporte de tradução entre as representações.

Krynski [35] apresenta um trabalho com uso de múltiplas representações que utiliza o DEfT como ponto de apoio. O trabalho se trata de uma ferramenta protótipo para o ensino de programação de computadores. O ambiente interativo apresenta simultaneamente ao aprendiz um código fonte em linguagem Pascal e o correspondente fluxograma (uma forma de grafo) que descreve a lógica de execução do programa. A ferramenta permite ao aprendiz montar um fluxograma que é traduzido simultaneamente para a representação em código fonte, assim o aprendiz consegue visualizar cada componente do fluxograma e a sua correspondência no texto do programa. O ambiente favorece o ensino da lógica e a familiarização com os elementos sintáticos de uma linguagem, demonstrando os benefícios do aprendizado através de MRE.

Para Spiro [73] no aprendizado em domínios mal estruturados, e mais especificamente na aquisição de conhecimento avançado, é fundamental a utilização de MRE em função de duas características que ocorrem: 1) cada caso ou exemplo de aplicação estudado envolve muitas estruturas conceituais complexas e 2) o padrão de incidência de conceitos pode variar substancialmente entre um caso e outro ainda que sejam nominalmente do mesmo tipo. A complexidade dos casos requer que sejam representados sob múltiplas perspectivas.

Ainsworth [1] afirma que em casos onde a carga de informações complexas diferentes é grande, o uso de MREs simplifica cada representação, tornando-as mais úteis. Também é útil nos casos onde exige-se um aprendizado mais profundo (avançado), com o fim de promover abstração, generalização e associação de representações.

2.2 Aprendizado Baseado em Jogos Digitais

Esta seção apresenta alguns estudos sobre a utilização de jogos digitais na educação e seus benefícios.

O termo "Aprendizado Baseado em Jogos Digitais" se tornou conhecido a partir da publicação do livro *Digital Game Based Learning* de Marc Prensky [57] e tem sido usado como referência sobre o tema. Entretanto ele não foi o primeiro autor a defender a ideia que jogos digitais são um excelente instrumento para o aprendizado. Em 1980, Malone [42] já traçava algumas ideias sobre jogos digitais no ensino e como eles deveriam ser projetados.

2.2.1 O problema da motivação

É evidente e sabido que o processo educativo (ensino-aprendizado) muitas vezes é tedioso e cansativo. Por esta razão existem muitas teorias e estudos, na área de psicologia educacional, que tratam sobre o problema de como melhorar a motivação e o engajamento dos estudantes no processo. Alguns exemplos destes estudos podem ser vistos em [19] e [89].

Os estudos citados no parágrafo anterior abordam o problema da motivação através de duas teorias da psicologia. O primeiro aborda auto-determinação e o segundo aborda auto-eficácia.

Auto-eficácia é um conceito que define uma forma de medir a motivação e o aprendizado de estudantes. Auto-determinação, quando aplicada à educação, diz respeito a como promover o interesse pelo aprendizado, valorar a educação e melhorar a confiança em suas próprias capacidades e atributos. Em outras palavras melhorar a auto-eficácia do aprendizado.

Deci [19] sugere que a motivação e a valoração da educação vem como resultado de um processo de internalização e integração. Ambas requerem que estudantes sejam capazes de se sentirem competentes, relacionados com o ambiente e autônomos enquanto resolvem atividades. Na sequência será abordado o uso de jogos e computadores na educação como forma de produzir tais resultados nos estudantes.

O termo jogo tem muitas definições segundo o dicionário Michaelis [46], dentre elas destacam-se três:

1. Qualquer atividade recreativa que tem por finalidade entreter, divertir ou distrair, brincadeira, entretenimento, folguedo.
2. Divertimento ou exercício de crianças em que elas demonstram sua habilidade, destreza ou astúcia.
3. Atividade, quando diferentes indivíduos ou grupos de indivíduos se submetem a competições em que um conjunto de regras determina quem ganha ou perde.

Para Huizinga [31], um jogo é uma atividade voluntária, exercida dentro de determinados limites de tempo e espaço, segundo regras livremente consentidas, mas absolutamente obrigatórias, dotado de um fim em si mesmo, acompanhado de um sentimento de tensão e alegria e de uma consciência de ser diferente da “vida cotidiana”. Esta definição é mais apropriada para esta tese, pois inclui a noção de limites temporais e espaciais. Um HLP também ocorre dentro destas limitações. Existe um instante onde ele surge, ocupa um espaço físico limitado e ocorre durante um intervalo de tempo limitado.

Computadores e dispositivos eletrônicos digitais como tablets, celulares inteligentes entre outros, são atrativos devido a sua grande versatilidade. São máquinas que atendem a múltiplos propósitos.

Considerando tanto o aspecto divertido dos jogos quanto a multifuncionalidade dos computadores, esforços têm sido feitos para o desenvolvimento de conceitos e ferramentas de integração de ambos de forma coesa ao processo educativo.

Malone [42] apresenta alguns elementos importantes que um jogo deve ter para ser introduzido como meio de aprendizagem:

- Regras;
- Objetivos claros e desafiadores;
- Níveis progressivos de dificuldade;
- Interação e alto grau de controle pelo estudante;
- Retorno (*feedback*) construtivo.

O uso de jogos digitais com estes elementos tem o potencial de ser útil tanto para a aquisição de conceitos quanto de habilidades por tornarem o processo mais motivante. A seguir são apresentados dois exemplos de sucesso do uso de jogos para fins educativos.

2.2.2 Aquisição de Conceitos e Habilidades

Em [55], foi apresentado um estudo comparativo entre dois grupos de estudantes. O objetivo dos dois grupos era aprender conceitos sobre memória de computadores. O primeiro grupo fez uso de um jogo desenvolvido, seguindo as sugestões de Malone. O segundo grupo utilizou um aplicativo baseado na *web*.

Os resultados indicaram que o grupo que fez uso do jogo foi mais efetivo na retenção dos conhecimentos do que o segundo grupo. Também foi constatado que a motivação era maior no primeiro grupo.

Virvou [82] apresenta um STI denominado VR-ENGAGE, que se utiliza de um jogo baseado em realidade virtual para o ensino de geografia. Seu estudo demonstrou que a utilização do jogo aumentou a motivação dos estudantes, diminuindo o índice de erros nos testes realizados. Três grupos distintos de estudantes foram avaliados no experimento: (1) desempenho fraco, (2) desempenho medíocre e (3) estudantes de bom desempenho. Os grupos (1) e (2) apresentaram significativa melhora nos resultados.

Andreu [4] também aponta que a aquisição de habilidades pressupõe um treinamento em situações de mundo real, através da prática de exercícios e da busca de conceitos aprendidos que devem ser corretamente utilizados, a este tipo de treinamento dá o nome de aprendizado ativo. Algumas técnicas de aprendizado ativo são: aprendizado baseado em solução de problemas, projetos, estudos de casos, simulações e jogos.

Simulações são especialmente úteis para este fim, pois apresentam uma modelagem do mundo real e, de acordo com Squire [74], permitem aos estudantes experimentar seus conhecimentos sob diversos aspectos, como:

- Manipulação de variáveis que de outra forma seriam inalteráveis (e.g volume de CO_2 na atmosfera);
- Visualização de fenômenos sob diversas perspectivas. Este seria o caso de simuladores empresariais, onde uma pessoa pode assumir a gerência de empresas concorrentes ou diferentes;
- Observação do comportamento de um sistema ao longo do tempo (reflexão);
- Formulação de questões ou situações hipotéticas, em outras palavras testar hipóteses;
- Visualização tridimensional do ambiente;
- Comparação entre o sistema simulado e seu entendimento do mundo real em questão (reflexão).

Na área da saúde, por exemplo, um estudo apresentado em [69] confirma que simulações facilitam o aprendizado experiencial. O estudo apresenta como ocorre a reflexão-nação utilizando um simulador denominado de *VR-Simulator*. O estudo foi realizado com alunos de odontologia e enfermagem, os quais precisam adquirir habilidades para perceber relacionamentos espaciais em imagens radiológicas. Esse treinamento normalmente se dá pela análise de imagens pré-produzidas para evitar riscos de exposição à radiação.

A figura 2.1 apresenta uma seção de treinamento com o simulador. Os alunos trabalham de forma colaborativa em cada seção.



Figura 2.1: Treinamento usando o VR-Simulator

Fonte: [69]

Pela exposição acima, percebe-se que o ensino das mais diversas áreas de conhecimento se beneficiam do uso de jogos digitais especialmente elaborados para este fim.

2.3 Xadrez, Informática e Educação

Jogos clássicos de estratégia e raciocínio, como o Xadrez, já eram utilizados na educação, mesmo antes do advento dos computadores, para transmitir conceitos geométricos, matemáticos, espaciais e habilidades de memorização e concentração, entre outras. Na próxima seção, serão vistos alguns trabalhos sobre a utilização do jogo de xadrez na educação.

Uma rápida busca no motor de busca acadêmico *scholar.google.com.br* torna latente como o xadrez tem sido alvo de pesquisas nos mais diversos níveis de escolaridade. Muitos estudos empíricos tem procurado demonstrar que sua inclusão nos currículos escolares apresenta resultados positivos.

Gobet [26], psicólogo cognitivista, foi o primeiro a escrever uma revisão de caráter mais crítico sobre os principais estudos existentes até então que tratam dos benefícios do xadrez na educação. Ele argumenta que boa parte dos estudos demandam melhor metodologia e suporte empírico, e propõe uma forma de experimento ideal para melhorar a confiabilidade e interpretação dos resultados.

Apesar das falhas metodológicas destes estudos, são um importante referencial para a defesa do ensino de xadrez e sua inclusão nas atividades escolares. Por isso é importante avançar nas pesquisas e no desenvolvimento de novas ferramentas de software voltadas para o ensino/aprendizagem. De fato, são poucos os trabalhos que tem este objetivo. As subseções seguintes apresentam dois destes trabalhos publicados, bem como demais artigos relacionados ao objeto deste trabalho: formalização de estratégias/planos de longo prazo.

2.3.1 Trabalhos correlatos

Em [25] é apresentado UMRAO, um STI para ensinar finais de Bispo-Peão. O sistema possui dois módulos principais: a) o módulo especialista e b) o módulo do tutor. O módulo especialista contém dois tipos de representações de conhecimento: a) planos estratégicos codificados por um humano e b) planos táticos (baseado em grafos) gerados de forma automática. Os planos possuem propriedades denominadas de *slots* que permitem especificar dentre outras coisas: qual lado vai jogar, tipo do plano (se é pra especialistas ou iniciantes), aplicabilidade, factibilidade, melhores metas, metas alternativas e restrições.

Cada propriedade é definida como uma conjunção de características presentes num dado posicionamento do tabuleiro, podendo incluir referências a outros planos previamente definidos. O módulo especialista produz um grafo com todos os possíveis movimentos, justificados pelos planos armazenados, cujas propriedades de aplicabilidade e factibilidade combinam com a configuração do tabuleiro dado.

A partida tem início com o aprendiz tentando mover uma peça, se o movimento que ele faz não combina com os movimentos pré-calculados pelo módulo especialista, a jogada deve ser refeita. O jogo continua com o STI fornecendo dicas das jogadas mais adequadas em cada situação. A figura 2.2 mostra exemplos do formalismo adotado para representar os planos. Cabe dizer que estes planos são de natureza interna do sistema, criados por especialistas, e que o aprendiz não tem acesso aos mesmos para modificá-los ou mesmo para criar os seus próprios.

Outro trabalho destinado ao ensino de xadrez foi realizado por Lazzeri [37]. O objetivo principal do trabalho é o ensino de táticas de meio de jogo. Para isto foi desenvolvido um STI denominado ICONCHESS (INteractive CONSultant for CHESS Middlegames).

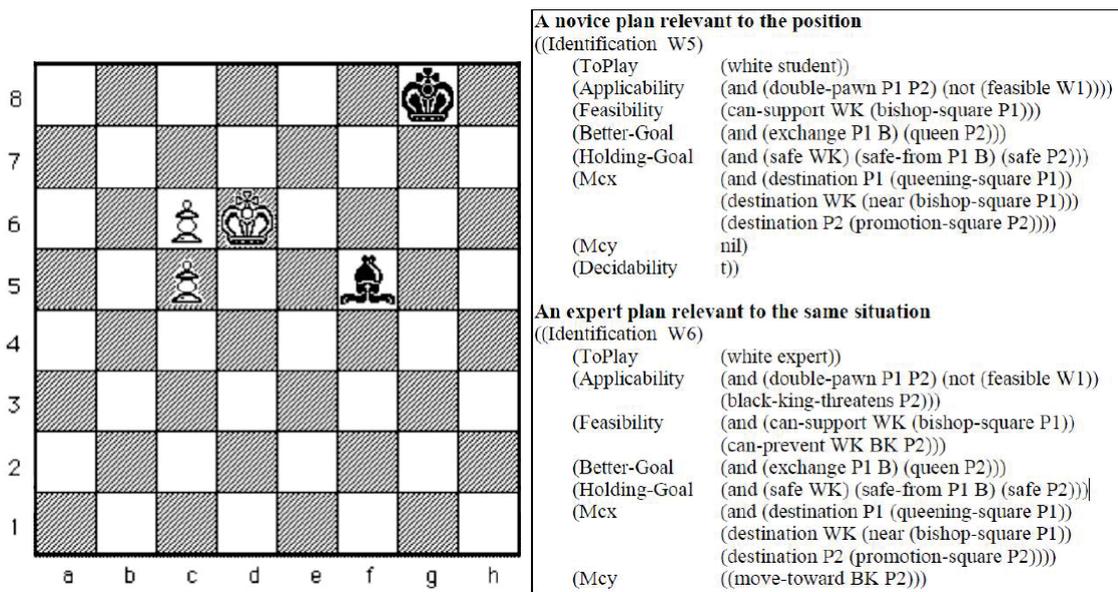


Figura 2.2: UMRAO - Codificação de planos

Fonte: [25]

O sistema solicita que o aluno forneça um caso próprio de meio de jogo e sugere opções de jogadas e análises de posicionamento. Para isto ele realiza uma busca por situações similares em suas bases internas de casos armazenados.

Organizado em dois módulos (Especialista, Tutor), o programa dispõe de três bases de conhecimento previamente montadas que são utilizadas para gerar as análises de jogadas: base de casos, base de regras e tabelas *Fuzzy*. As tabelas *Fuzzy* são usadas para calcular quão similar a estrutura de peões do tabuleiro fornecido está em relação aos casos armazenados na base. Cada tabela representa uma estrutura canônica de peões. Uma vez selecionados os casos parecidos, o aprendiz pode comparar os casos e solicitar dicas sobre cada um deles.

Apesar do ICONCHESS ser considerado um STI, os resultados do estudo apontaram que aprendizes iniciantes e médios se beneficiaram com o sistema, ajudando a compreender melhor conceitos como mobilidade e posicionamento de peças. Já os mais experientes tiveram pouco ganho em seu aprendizado. Talvez a falta de um módulo de acompanhamento da evolução do aprendiz tenha contribuído para tal [37].

2.3.2 IA aplicada a jogos de tabuleiro

Jogos de tabuleiro como damas, jogo da velha e xadrez recaem no campo da resolução de problemas por meio de busca. Newell [52] foi o primeiro a formular um modelo de espaço de estados na teoria de resolução de problemas. Um espaço-problema ou espaço de estados é definido como:

- Um conjunto de estados;
- Um conjunto de operadores que conectam um ou mais estados entre si. Os operadores permitem gerar transições entre estados onde certas condições sejam aplicáveis.

Assim, um problema é composto de um espaço-problema, um estado inicial e um conjunto de estados objetivo ou solução. A tarefa se trata de encontrar uma sequência de operadores

(transições) que liguem o estado inicial até um estado solução. Três fatores devem ser considerados na busca: tempo e tamanho do espaço necessários para encontrar a solução e o custo da solução em si [33].

Diversos algoritmos ou abordagens foram desenvolvidos para realizar esta tarefa, e de acordo com Korf [33] podem ser classificados em:

- Força bruta: neste tipo de algoritmo não existe conhecimento ou informação aplicado ao processo de busca. Dado o estado o inicial, geram-se todos os estados possíveis a partir deste verificando se algum deles pertence ao conjunto solução. Caso negativo, para cada estado gerado, geram-se os novos estados possíveis e assim sucessivamente até encontrar um estado solução ou não ser mais possível gerar novos estados. Sua complexidade depende de dois parâmetros do problema: ramificação e profundidade. Ramificação é a média de novos estados gerados a partir de um estado qualquer e a profundidade é o tamanho do menor caminho entre o estado inicial e um estado solução. A busca pode se dar tanto em largura quanto em profundidade.
- Busca heurística: acopla-se ao método da força bruta uma função de avaliação estática que mantém conhecimento especializado sobre o domínio. Cada estado é avaliado por esta função com a finalidade de estimar o custo para atingir um estado-objetivo. Esta é uma forma de reduzir o espaço de busca, eliminando os caminhos mais improváveis (e.g MINIMAX com podas alfa-beta, A^* , *Hill-climbing*). Diversos avanços foram realizados para torná-la mais eficiente, como por exemplo: busca com profundidade iterativa (para jogadas com limitação de tempo) [33], janelas de aspiração [61], livros de abertura [44] entre outros;
- Subobjetivos: a ideia é quebrar o problema em partes menores a fim de diminuir o custo da busca.

Korf [33] define três categorias de subobjetivos: (1) Independentes, (2) serializáveis e (3) não serializáveis:

Subobjetivos são independentes se cada operador diminui a distância apenas em relação a um subobjetivo. Assim os operadores podem ser divididos em subconjuntos onde os operadores de cada um deles são relevantes apenas para um subobjetivo singular. Tal estratégia reduz o fator de ramificação bem como a complexidade de tempo para a execução da tarefa.

Korf [33] discute também a ideia de subobjetivos serializáveis como uma forma de tratar o espaço de estados, tal ideia foi apresentada inicialmente por Newell [52]. Um conjunto de subobjetivos é serializável se existe uma ordem entre seus elementos de maneira que possam ser resolvidos sequencialmente sem interferir nos resultados já obtidos. A vantagem de buscar uma serialização de subobjetivos é diminuir o fator de ramificação de uma árvore de estados. Movimentos que violam a solução já obtida são descartados, diminuindo o tamanho da busca.

Subobjetivos podem ser não serializáveis. Isto ocorre quando uma solução parcial de um problema já alcançado precisa ser modificada para avançar na solução global do problema. Uma forma de tratar seria utilizando macro-operadores para alcançar o próximo estado de solução parcial. Macro-operadores são sequências de operadores primitivos e seriam úteis quando eles modificam as soluções já alcançadas, mas apenas temporariamente, retornando-as ao seu estado original [33].

Comparando a taxonomia apresentada por Kotov (1989,p.82-83 *apud* [15]) sobre

planos que enxadristas estabelecem durante uma partida de xadrez, podendo ser mono-escalonados ou multi-escalonados com a classificação de subobjetivos dada por Korf [33] pode-se fazer uma correlação entre os planos mono-escalonados com os subobjetivos independentes, e os multi-escalonados com os serializáveis. Assim é adequado sugerir que estabelecer planos com subobjetivos independentes e/ou serializáveis pode ser uma maneira promissora para construção de máquinas com características estratégicas.

Pachman [54] afirma que um jogador de xadrez estrategista possui a habilidade de conduzir um jogo inteiro de acordo com um plano, chamado de plano estratégico. Um plano poderia ser alcançar um determinado padrão de posicionamento que apresente uma ligeira vantagem (e.g. controle de uma fileira, conseguir um peão passado, ou enfraquecer um peão adversário). Assim é razoável afirmar que a técnica de busca por subobjetivos está implícita na busca que um estrategista faz por vantagens posicionais que alterem o balanço de forças de jogo em seu favor.

A seção seguinte aborda alguns conceitos na literatura de gestão empresarial, bem como na literatura de aplicação do domínio deste trabalho, sobre estratégia e planejamento estratégico.

2.4 Estratégia e ferramentas de planejamento

Beinhocker [5] apresenta algumas definições sobre estratégia que abstraem elementos importantes, os quais devem ser considerados na formalização de estratégias, seja em um jogador automático, ou numa ferramenta voltada para o ensino de tais conceitos:

p.324 "Estratégia pode ser definida como a determinação de objetivos e metas de longo prazo básicos de um empreendimento, e a adoção de ação e a alocação de recursos necessários para consecução destes objetivos... Primeiro, estratégia é inerente a olhar para frente. Para desenvolver uma estratégia, alguém tem que fazer uma determinação sobre onde deseja estar no futuro. Segundo, estratégia, diz respeito a criar um plano para chegar ao estado futuro desejado e comprometer-se com um curso de ação definida por este plano" - traduzido pelo autor.

p.334 "Ao invés de pensar sobre estratégia como um simples plano construído sobre predições do futuro, deveríamos pensar sobre estratégia como um portfólio de experimentos, um conjunto de planos...competindo que evoluem ao longo do tempo" - traduzido pelo autor.

p.337 "Construir um portfólio de experimentos **requer um entendimento coletivo da situação corrente e das aspirações partilhadas**...processos precisam ser estabelecidos que permitam a amplificação dos Planos bem sucedidos... e a eliminação de planos mau sucedidos." - traduzido e grifado pelo autor.

Em ciências humanas, como administração e economia, planejamento estratégico é uma disciplina bastante estudada. Diversas técnicas foram desenvolvidos para criar e avaliar estratégias de negócio, tais como: COA (*Course of Actions*) [30], matriz de mercado/produto, Porter (análise competitiva), análise de portfólio, análise SWOT (*Strengths, Weaknesses, Opportunities, Threats*), entre outras [84].

A análise COA, por exemplo, já foi objeto de estudo científico para tratamento de incertezas em jogos de guerra táticos [30]. Outra técnica promissora para jogos como

o xadrez, em especial no aspecto de ensino, é a técnica SWOT. Esta técnica permite uma reflexão sistemática sobre a situação atual a fim de identificar pontos fortes, fracos, oportunidades e ameaças (riscos) [85].

Segundo Weihrich [85] a análise SWOT permite estabelecer quatro abordagens estratégicas com objetivos específicos:

- Estratégia 1: análise WT (mini-mini). Nesta abordagem o objetivo é minimizar as fraquezas e as ameaças.
- Estratégia 2: análise WO (mini-maxi). Minimizar fraquezas e maximizar as oportunidades.
- Estratégia 3: análise ST (maxi-mini). Maximizar os pontos fortes e minimizar as ameaças.
- Estratégia 4: análise SO (maxi-maxi). Maximizar os pontos fortes e oportunidades.

As quatro avaliações acima são denominadas de matriz de relacionamento, onde procura-se verificar quais elementos de um combinam, influenciam ou têm relação com os elementos de outro. A partir desta análise o estrategista estabelece cenários futuros e define quais táticas/ações serão necessárias para alcançá-los no mundo real.

Algumas iniciativas particulares, sem caráter científico, propõe o uso desta técnica no xadrez, como pode ser visto em <https://www.ichess.net/2011/06/03/chess-in-business-improvement-swot-analysis/> (acesso em 08/12/2018). Outro exemplo encontra-se em <http://chesshound.blogspot.com.br/2011/03/swot-evaluation-and-analysis-in-chess.html> - acesso em 08/12/2018.

Na próxima subseção serão revistos artigos que abordam o estabelecimento de planos estratégicos no xadrez em jogadores automáticos. Tal revisão é importante para embasar modelos de formalismo, padrões e conceitos que possam ser utilizados no presente trabalho.

2.5 Jogadores automáticos e planejamento estratégico

Poucos trabalhos apresentaram modelos e experimentos na criação de jogadores automáticos que utilizam conhecimento, padrões e construção de planos a fim de selecionar bons ou ótimos movimentos. Importante ressaltar que nenhum destes trabalhos foi idealizado para uso em atividades de ensino de xadrez.

Pitrat [56] fez um programa que analisava certa posição de meio de jogo, criava alguns planos e tentava executá-los enquanto considerava ramificações da árvore de busca, descritas como sendo "ramificações naturais". Algumas regras do xadrez foram ignoradas no trabalho para simplificar o processo: roques, *en passant* e afogamento do rei. O programa recebia como entrada uma posição de tabuleiro e um valor de combinação (balanço material) que ele precisava encontrar. O programa também "sabia" que tinha que encontrar um xeque-mate, capturar uma torre ou apenas um peão.

Para descrever os planos, Pitrat [56] propôs uma linguagem com quatro tipos de sentenças. A tabela 2.1 apresenta os elementos desta linguagem.

Um estado é representado da seguinte forma no modelo proposto por Pitrat [56]: $[N]EM/FM/V$. Onde N representa a negação do estado, E representa peça inimiga, F representa peça do jogador, V representa casa vazia, e M representa o tipo da peça.

Tabela 2.1: Linguagem para construção de planos - Pitrat

Tipo de sentença	Sintaxe	Descrição
Movimento de peças	$xyz t^*$	x, z representam as linhas pelos caracteres de 'a'..'h'; y, t representam as colunas de 1 a 8; * representa um movimento que causa uma ameaça para o oponente
Sentença de alteração	$xy o \rightarrow n$	xy é a coordenada da casa a ser modificada o representa o estado atual da casa n o novo estado desejado para a casa
Sentença de verificação	$xy e$	xy é a coordenada da casa a ser verificada e é o estado da casa que se quer verificar
Sentença de análise	$xy analysis$	xy é a coordenada da casa, onde está uma peça inimiga, a ser analisado seguido da palavra <i>analysis</i> . Busca por outras combinações mais simples como ataques duplos, por exemplo.

A listagem 2.1 apresenta uma sequência de planos/sub-planos idealizados para a situação de tabuleiro da figura 2.3, cujo objetivo é capturar um cavalo das peças brancas.

1	P1. Remove the black knight from d5
2	d8d1
3	d1g1
4	P2. d5c3
5	Verify that d5 is empty
6	d8d1
7	d1g1
8	P3. Remove the white rook from c1
9	d8d1
10	d1g1
11	P4. c5c3
12	Verify that there is no white rook in c1.
13	d8d1
14	d1g1
15	P5. Remove the white queen from e2
16	d8d1
17	d1g1

Listagem 2.1: Exemplo nível 04 - transição de etapa - Fonte: [56]

A linguagem proposta por Pitrat [56] ficou bastante restrita se limitando apenas a quatro primitivas e representa apenas o processo de execução de um plano. Entretanto foi um importante passo para usar conhecimento ao invés de força bruta.

Wilkins [87] apresenta um trabalho mais complexo, pois além da possibilidade de criar planos através de uma linguagem de padrões, o sistema desenvolvido possuía ainda uma base de conhecimentos que era utilizada para conduzir a busca. Como fator limitador, também era um sistema voltado para posições de meio de jogo, embora o autor defendia que poderia ser usado em outras etapas.

O sistema foi denominado como PARADISE (*Pattern Recognition Applied to Directing Search*). Baseado em um conjunto de 200 regras de produção, nas quais a condição era formada por um conjunto de padrões (um conjunto complexo de características) e a conclusão era composta de um ou mais conceitos que eram utilizados para definir um plano de ação. Os conceitos eram introduzidos numa base de fatos sobre o tabuleiro.

A listagem 2.2 permite reconhecer peças do oponente que se encontram presas, sem

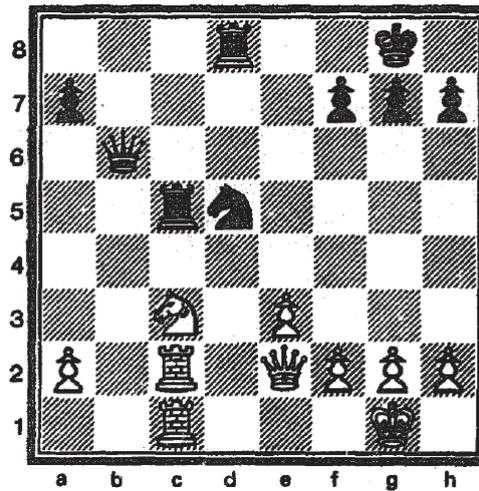


Figura 2.3: Combinação de peças encontrada para aplicação de planos

Fonte: [56]

opções de movimento. A primeira linha é uma variável que é instanciada para cada peça do oponente. A segunda linha verifica se esta peça não pode mover-se sem ser capturada. A terceira linha verifica se a peça não se encontra em posição de ser capturada. Caso a regra seja válida uma ação é conduzida, que é colocar na base de fatos um conceito de ataque a esta peça fragilizada.

```

1 ((DMP1)
2 (NEVER (EXISTS (SO) (PATTERN MOBIL DMP1 SO)))
3 (NEVER (EXISTS (P1) (PATTERN ENPRIS P1 DMP1)))
4 (ACTION ATTACK ((OTHER-COLOR DMP1) (LOCATION DMP1)) (THREAT (WIN DMP1))
5 (LIKELY 0)))

```

Listagem 2.2: Regra de produção para peças presas. Fonte: [87]

A figura 2.4 mostra uma situação de jogo que é analisada por PARADISE através de um módulo de análise estática, com base nas regras de produção. Essa análise produz planos. A listagem 2.3 contém uma instanciação de um plano. O plano inicia com um movimento e segue fazendo verificações de movimentos do oponente e gerando movimentos para cada situação esperada. A intenção é capturar o Rei preto, mas alternativamente pode-se capturar a Dama preta ou uma Torre preta.

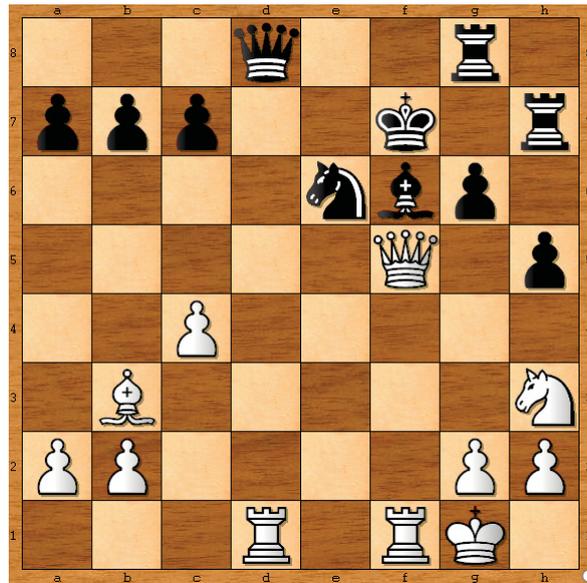


Figura 2.4: Posição analisada - brancas jogam

```

1  (((WN N5)
2  (((BN N4) (SAFEMOVE WR 07)
3  (((BK NIL) (SAFECAPTURE WR BR))
4  ((ANYBUT BK) (SAFECAPTURE WR BK))))
5  ((BN N4) (CHECKMOVE WR 07) (BK NIL) (SAFECAPTURE WR BQ) )))
6  ((THREAT (PLUS (EXCHVAL WN N5) (FORK WR BK BR)))
7  (LIKELY 0))
8  ((THREAT (PLUS (EXCHVAL WN N5) (EXCH WR BQ)))
9  (LIKELY 0)))

```

Listagem 2.3: Regra de produção para peças presas instanciada. Fonte: [87]

Schaeffer [60] apresenta um jogador automático denominado *Planner* para posições de meio de jogo, composto por três módulos principais: 1) seletor de planos, 2) estratégia e 3) gerador de movimentos. O seletor de planos, avalia a posição atual e escolhe um plano de ação. O módulo de estratégias identifica o que precisa ser feito e estabelece um posicionamento de peças desejável para atingir o objetivo. O gerador de movimentos recebe do módulo de estratégia a lista de peças e posições, e uma pontuação para colocar as peças nestas posições.

O sistema de Schaeffer [60] se baseia em um conjunto com 13 planos pré-construídos. Os planos são escolhidos depois de extensa e complexa análise da posição atual. Cada plano possui um modelo com várias características que se estiverem presentes, conduzem à escolha do respectivo plano como candidato. Probabilidade, factibilidade e praticidade são calculadas para cada plano candidato e o de mais alta pontuação nestes critérios é escolhido, entretanto o artigo não deixa claro como isto é feito.

Kaindl [32] estabelece uma busca por padrões posicionais conhecidos onde podem ser aplicadas estratégias de longo-prazo. Uma vez identificado um padrão posicional, estabelece-se uma sequência de posições futuras relativas ao padrão inicial (subobjetivos) e um padrão final como objetivo principal. Cada subobjetivo recebe um valor que identifica a porcentagem de aproximação do padrão final. As posições são descritas através de uma linguagem livre de contexto criada para representar os padrões posicionais. Trata-se de uma linguagem descritiva e é limitada pelo fato que o padrão inicial deve ser baseado apenas em posicionamento de peões.

O plano é descrito inicialmente por um nome, e duas linhas descrevendo quais fileiras interessam para cada cor de peça. As demais fileiras são ignoradas. Na sequência,

os padrões são descritos em três partes: MPP (padrão inicial), CPP (padrão condicional) e opcionalmente NPPP (padrão para outras peças).

A geração de padrões é feita por um programa especialmente projetados para isto. Os padrões são representados internamente por tabelas *hash*. Uma interface foi criada para conectar a um programa jogador de xadrez de código aberto denominado MERLIN. A função de avaliação estática do MERLIN foi modificada para determinar se os peões encontrados nas fileiras correspondem a algum padrão da tabela *hash*. A listagem 2.4 apresenta um exemplo dos componentes da linguagem de padrões.

```

1 MINORITYATTACK:
2
3 KPS L A-FILE, B-FILE, C-FILE, D-FILE E-FILE
4   D A-FILE, B-FILE, C-FILE, D-FILE E-FILE
5
6 MPP L B2, NOPAWN C-FILE, D4
7   D B7, C6           : +20
8
9 CPP IF COUNTERPAWN(A-FILE) THEN
10    IF RAM(A4, A5) THEN : -100
11    ELSE IF D A5 THEN
12        IF L A3 THEN : -5
13        ELSE : -20
14        ENDIF
15    ELSE : 0
16    ENDIF
17  ENDIF
18  ELSE IF EMPTY(A-FILE) THEN : -5
19  ENDIF,
20
21  IF RAM(D4, D5) THEN : 0
22  ELSE IF COUNTERPAWN(E-FILE) AND D E4 THEN : -35
23  ENDIF
24
25  ENDIF

```

Listagem 2.4: Linguagem de padrões de peões. Fonte: [32]

Uma das características que difere os algoritmos jogadores automáticos e os jogadores humanos, é que estes são mais seletivos nas suas escolhas de movimentos. Apenas os movimentos mais promissores são analisados e os demais são descartados. Tendo isto como base, Finkelstein [24] propôs uma metodologia, no campo do aprendizado de máquina, baseada em padrões de posicionamento e movimento relativos entre as peças, incluindo a resposta do oponente e os pesos referentes a trocas materiais.

Os padrões são representados através de grafos acíclicos e podem ser estáticos ou dinâmicos, veja a figura 2.5, onde F representa uma peça e S representa um quadrado no tabuleiro. A figura apresenta os tipos de relações dinâmicas em forma de padrões de posicionamento e movimentação. A vantagem desta abordagem é a expressividade semântica do padrão, o que torna possível expressar em linguagem natural. Entretanto faltam representações mais compactas com maior semântica, por exemplo domínio de fileiras, linhas e diagonais. Outro aspecto não deixado claro é um ataque múltiplo, conhecido como garfo, utilizado como ilustração no artigo, para o qual não foi definido o tipo de padrão dinâmico.

Outro trabalho no campo do aprendizado de máquina, aplicado ao domínio de xadrez, foi realizado por Tadepalli [78]. O método denominado de *Lazy Explanation-Based Learning* (LEBL) é uma evolução dos trabalhos de Mitchell [49] e Minton [48]. Um sistema de aprendizado EBL (*Explanation-Based Learning*) refere-se a capacidade que um sistema possui de explicar a si próprio, baseado nas teorias do domínio, que um exemplo de treinamento é um membro do conceito alvo.

O sistema proposto por Tadepalli [78] produz explicações incompletas e genéricas dos casos de treinamento, produzindo como resultado o que são chamados de planos otimistas. O ponto de interesse nesse trabalho recai sobre o modelo de representação do conhecimento proposto para formalizar os planos. A figura 2.7 apresenta um formalismo onde o objetivo, dada a situação de jogo apresentada na figura 2.6, é coroar o peão.

Edge type	Arity	Meaning	Example
FS	2	F controls square S	
F_1F_2	2	F_1 directly attacks/defends F_2	
F_1SF_2	3	F_1 controls square S , from where it would attack/defend directly F_2	
$F_1F_2F_3$	3	F_1 attacks/defends F_3 indirectly through F_2	
$F_1SF_2F_3$	4	F_1 can move to square S , from where it will attack/defend F_3 indirectly through F_2	
$F_1F_2SF_3$	4	F_1 directly attacks/defends F_3 , and F_2 can be wedged between them on square S	
$F_1F_2F_3S$	4	F_3 can move to square S , from where it will be attacked/defended by F_1 indirectly through F_2	
$F_1F_2F_3F_4$	4	F_1 attacks/defends F_4 indirectly through F_2 and F_3	
$F_1F_2SF_3F_4$	5	F_1 attacks/defends F_4 indirectly through F_3 , and F_2 can be wedged between F_1 and F_3 on square S	
$F_1F_2F_3SF_4$	5	F_1 attacks/defends F_4 indirectly through F_2 , and F_3 can be wedged between F_2 and F_4 on square S	

Figura 2.5: Representação de padrões dinâmicos - Fonte: [24]

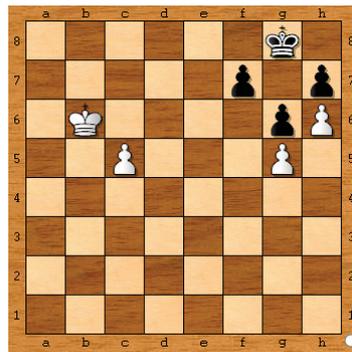


Figura 2.6: Estudo de caso - Fonte: [78]

```

Player: WHITE
Goal: QUEEN-WHITE-PAWN
Body: [If((ON ?p ?x 5) (TYPE ?p PAWN)
  (OWNS WHITE ?p) (FREE ?x 6)
  (FREE ?x 7) (FREE ?x 8))
  Then PAWN-MOVE(WHITE ?x 5 ?x 6)
  If((ON ?p ?x 6) (TYPE ?p PAWN)
  (OWNS WHITE ?p) (FREE ?x 7) (FREE ?x 8))
  Then PAWN-MOVE(WHITE ?x 6 ?x 7)
  If((ON ?p ?x 7) (TYPE ?p PAWN)
  (OWNS WHITE ?p) (FREE ?x 8))
  Then PAWN-MOVE(WHITE ?x 7 ?x 8)]

```

Figura 2.7: Plano otimista genérico - Fonte: [78]

O plano é definido para o jogador branco pela cláusula *Player*. O objetivo de coroar o peão é definido na cláusula *Goal*. A partir de então é detalhado o plano na cláusula *Body*, que para esta posição equivale a mover o peão para as casas c6, c7 e c8. A primeira instrução *if* verifica se existe um peão $?p$ na linha 5 em alguma fileira $?x$ e se o peão é branco e se as linhas 6, 7 e 8 estão livres na fileira $?x$. Caso verdadeiro o peão deverá realizar um movimento da linha 5 para a 7. A segunda instrução, semelhantemente faz a verificação a partir da linha 6. A terceira instrução verifica a partir da linha 7, finalizando o plano que permite coroar um peão. O plano é dito otimista pois considera

que o peão seja coroado, sem interposições de peças de longo alcance do oponente e nem mesmo a possibilidade de captura deste peão.

O aspecto positivo está na representação genérica de posicionamento de peças no tabuleiro, sem fixar uma região específica, além de ser um modelo mais simples que o proposto por Wilkins e de mais alto nível que o do Pitrat. Porém, se o peão estiver na linha 2 ou 3, o plano falha, ainda que o caminho esteja livre para o peão e o rei consiga dar suporte para o mesmo.

2.6 Trabalhos recentes

As principais pesquisas direcionadas para aplicar conhecimento que simule a forma de um grande mestre jogar, com o objetivo de minimizar a busca por força bruta, foram realizadas nas décadas de 70 e 80, até meados da década de 90, havendo uma perceptível diminuição a partir de então no número de artigos publicados.

Três hipóteses especulativas para tentar explicar tal fato seriam: (a) o surgimento de máquinas com mais capacidade de memória e processamento, e considerando que as técnicas de busca por força bruta estão bem entendidas e aprimoradas, reduziu a necessidade de pesquisa por métodos alternativos, (b) a derrota do grande mestre Kasparov pelo Deep Blue pode ter dado a impressão de "missão cumprida" por parte de muitos pesquisadores, e (c) na década de 90 a IA começou a expandir a área de testes/campos de prova para incluir situações do mundo real (e.g: otimização de entrega de produtos *just-in-time* com estoque mínimo, alocação de recursos humanos com locais geograficamente isolados e tempo de trabalhos variados).

Apesar da perceptível diminuição das pesquisas, outras abordagens surgiram na literatura, até o ano de 2010, como, por exemplo, as iniciativas deste departamento citadas na introdução deste trabalho, e também pesquisas no campo do aprendizado de máquina, redes neurais e algoritmos genéticos [36], [16]. No tema de busca seletiva encontram-se [59], [79] e [75].

Devido ao aumento da capacidade de processamento das máquinas atuais, abordagens utilizando redes neurais com aprendizado profundo (*deep learning*) em jogos adversaristas se tornaram viáveis. Assim, a comunidade científica retomou as pesquisas.

Um dos casos recentes foi o sucesso no jogo Go com o algoritmo *AlphaGo Zero*[62]. *AlphaGo Zero* foi capaz de ganhar dos principais jogadores mundiais, algo que não havia acontecido antes. Usando redes neurais convolucionais¹ e aprendizado com reforço jogando contra si mesmo.

Go possui uma explosão combinatória muito maior que o jogo de xadrez, da ordem de 10^{360} para um tabuleiro de 19x19 casas, porém as regras do jogo são mais simples, contendo apenas um tipo de peça para cada jogador e o objetivo é dominar mais áreas do tabuleiro. As peças de um jogador são capturadas quando se encontram cercadas pelas peças do oponente. Em seguida, o espaço liberado recebe novas peças do oponente.

Usando uma abordagem similar ao projeto *AlphaGo Zero*, porém com um algoritmo, segundo Silver *et. al.* [63], totalmente genérico denominado de *AlphaZero*, o qual foi aplicado em 3 jogos diferentes: Xadrez, Shogi e Go. Apenas aplicando o conhecimento das regras de cada jogo, sem nenhum conhecimento adicional e um treinamento da rede neural

¹Classe de rede neural do tipo *feed-forward*. Pertence ao campo da IA conexionista. O termo convolucional significa que os valores matemáticos calculados pela rede estão em constante transformação, a cada iteração no treinamento da rede

que durou três dias, uma disputa de 100 partidas foi realizada para cada tipo de jogo contra três máquinas. Duas delas bem conhecidas das suas comunidades: *Stockfish 8* para o Xadrez e *Elmo* para o jogo Shogi. Para o jogo Go o próprio *AlphaGo Zero* foi utilizado.

O algoritmo foi testado em 50 partidas como brancas e 50 como pretas. Pelos resultados apresentados em [63], *AlphaZero* alcançou o nível de mestre em cada um dos jogos. No caso específico de xadrez, o algoritmo avaliava 80 mil posições por segundo contra 70 milhões do *Stockfish*. Esta diferença enorme de posições avaliadas se deve à utilização do algoritmo *Monte-Carlo Tree Search*(MCTS), que utiliza uma abordagem de busca diferente do MINIMAX usado pela máquina *Stockfish*.

Este algoritmo apresentado pela primeira vez em 1993, [10] aplicado ao jogo Go, é uma adaptação do algoritmo de têmpera simulada para árvores de busca em jogos adversaristas. Mais detalhes sobre este algoritmo podem ser encontrados em [9].

AlphaZero, todavia por utilizar como algoritmo de busca o MCTS, não é um algoritmo que leva em conta uma estratégia, e tão pouco existe a percepção de padrões no nível de visão estratégica, pois a árvore é expandida até encontrar um xeque-mate ou empate. Além disto é necessário passar por uma fase de treinamento extensa para ajustar a rede neural.

Usando uma abordagem inovadora, Linhares [39] apresenta o projeto *Capyblanca* para abstração de papéis e visão estratégica no jogo de xadrez. Um algoritmo baseado em analogias como aspecto central à cognição. Utilizando uma arquitetura paralela de múltiplos níveis. Conforme descrito no artigo, dada uma posição qualquer, *Capyblanca* avalia peças encontradas em casas aleatórias e gradualmente faz uma avaliação sensível ao contexto do cenário como um todo.

Capyblanca não realiza buscas na árvore de jogo, porém avalia intensamente uma posição na tentativa de encontrar a essência do ponto de vista estratégico. O caminho adotado no *Capyblanca* é diferente da abordagem proposta nesta pesquisa, onde um dos objetivos é permitir que humanos formalizem seus conhecimentos heurísticos no nível de visão estratégica.

3 Materiais e métodos

O capítulo anterior apresentou uma revisão de literatura e contextualização de elementos teóricos relativos ao objeto desta pesquisa. Especial atenção foi dada aos trabalhos com formalismos para representar conhecimento do domínio.

Considerando, portanto, o que foi visto na literatura e o conceito de estratégia adotado na introdução, percebe-se uma lacuna a ser preenchida de conceitos e um método de formalismo para humanos explicitarem seu conhecimento de padrões no nível de visão estratégica. Tal conhecimento ficará armazenado em uma base de conhecimento e poderá ser usado tanto para busca em bases de jogos, ou para avaliar posições onde o conhecimento tático das máquinas atuais não avaliam corretamente e futuramente em uma máquina com visão estratégica.

Os seguintes passos metodológicos foram adotados como parte da solução do problema:

- Estudo de técnicas, tecnologias e linguagens para desenvolvimento da *arquitetura*;
- Definição de uma ontologia sobre padrões;
- Elaboração de uma linguagem para especificação de padrões de alto nível (HLPs);
- Especificação de ferramentas de *software* para autoria de HLPs;
- Especificação do analisador sintático e interpretador integrados às ferramentas;
- Realização de testes e experimentos, vide capítulo 5.

Estes passos metodológicos serão apresentados neste capítulo que encontra-se assim dividido: A seção 3.1 apresenta as tecnologias usadas; a seção 3.2 os passos metodológicos adotados para a solução do problema.

3.1 Tecnologias usadas

As tecnologias a seguir foram utilizadas para a execução deste trabalho:

- Netbeans 8.2: ambiente de desenvolvimento de código aberto distribuído por netbeans.org sob duas licenças: *Common Development and Distribution License (CDDL)* e a *GNU General Public License*.
- Java™ 1.8: linguagem multipropósito, multiplataforma e multi-dispositivos de propriedade da Oracle Corporation.
- Javacc™ 5.0: um gerador de *parsers* desenvolvido e distribuído por javacc.org.

- TuProlog 3.2.1: é uma máquina Prolog baseada em Java de código aberto.
- Protégé 5.5.0: ferramenta livre para definição de ontologias.
- Banco de dados: POSTGRESQL 9.3 (persistência);
- Máquina virtual "servheuchess" disponível nos computadores do DINF para os experimentos de busca em bases de jogos, contando com 12 núcleos de processamento.

A linguagem Java foi utilizada para:

- criação da ferramenta de autoria;
- programação dos conceitos e elementos condicionais dos padrões na forma de funções predefinidas;
- programação do avaliador sintático e interpretador;
- programação dos métodos de reconhecimento;
- interfaces de ativação de busca e visualização dos padrões usando MREs;

Javacc foi utilizado para a especificação das regras gramaticais da nova linguagem. Usa uma especificação similar à Backus-Naur-Form (BNF) para especificação de *tokens*¹ e regras que devem ser descritas num arquivo de extensão ".jj".

Javacc lê o arquivo ".jj" e produz como saída 7 outros arquivos com classes de objetos Java: Parser, ParserConstants, ParserTokenManager, ParserException, SimpleCharStream, Token e TokenMgrError. Dada uma gramática especificada num arquivo ".jj" e uma vez geradas estas classes, é possível então escrever códigos-fonte desta gramática em outro arquivo texto, avaliar a sua sintaxe através da classe Parser e preparar os códigos intermediários para execução ou interpretação.

TuProlog é uma máquina Prolog que possui como diferencial a capacidade de trabalhar com mais de uma base de fatos, que são chamadas de teorias, veja [20] para mais detalhes sobre o TuProlog. O Prolog, por sua vez, possui um algoritmo para casamento de padrões e foi utilizado no interpretador da linguagem PATLAN para verificar se um cenário encontra-se presente em uma posição de jogo.

3.2 Elementos da Solução

Esta seção apresenta os elementos criados para a solução do problema abordado por esta tese. A seção 3.2.1 apresenta uma ontologia para representar HLPs. A seção 3.2.2 apresenta a abordagem adotada para criação da linguagem PATLAN. A seção 3.2.3 apresenta as ferramentas que darão suporte ao uso da linguagem PATLAN e a realização dos experimentos. E por fim, a seção 3.2.4 descreve a metodologia adotada para os experimentos.

¹*Tokens* são os elementos simbólicos que fazem parte da gramática de uma linguagem, também chamados de unidades léxicas



Figura 3.1: Ontologia de HLPs - Diagrama de classes

3.2.1 Uma Ontologia para Representação de HLPs

Como primeiro elemento da solução do problema, após conversas informais com enxadristas, foi especificada uma ontologia para representar o conhecimento de HLPs. A terminologia adotada baseia-se em Linhares [40] e [24].

A figura 3.1 apresenta a hierarquia das classes da ontologia e a figura 3.2 mostra o diagrama de relacionamento entre as classes. A ontologia foi definida na língua inglesa para acesso da comunidade acadêmica internacional.

A classe HLP define a tipologia de padrões em duas subclasses: padrões ativos e padrões passivos. Padrões passivos não necessitam de um cenário, mas apenas da prova de conceitos na forma de condições iniciais (PreCondition). Padrões ativos possuem um cenário associado e se subdividem em duas classes: sequencia forçada de lances (o oponente não consegue interromper) e sequencia não forçada (o oponente pode interromper se tiver peças bem posicionadas no tabuleiro). O cenário por sua vez descreve ações que estão ocorrendo simultaneamente no objeto alvo da busca pelo padrão. As ações podem ser diretas (estáticas) ou indiretas (dinâmicas). Cada ação necessita de pelo menos dois atores participando podendo ser atores amigos ou inimigos. A ação também descreve um papel para cada ator participante. Os papéis podem ser: atacante, defensor, figurante

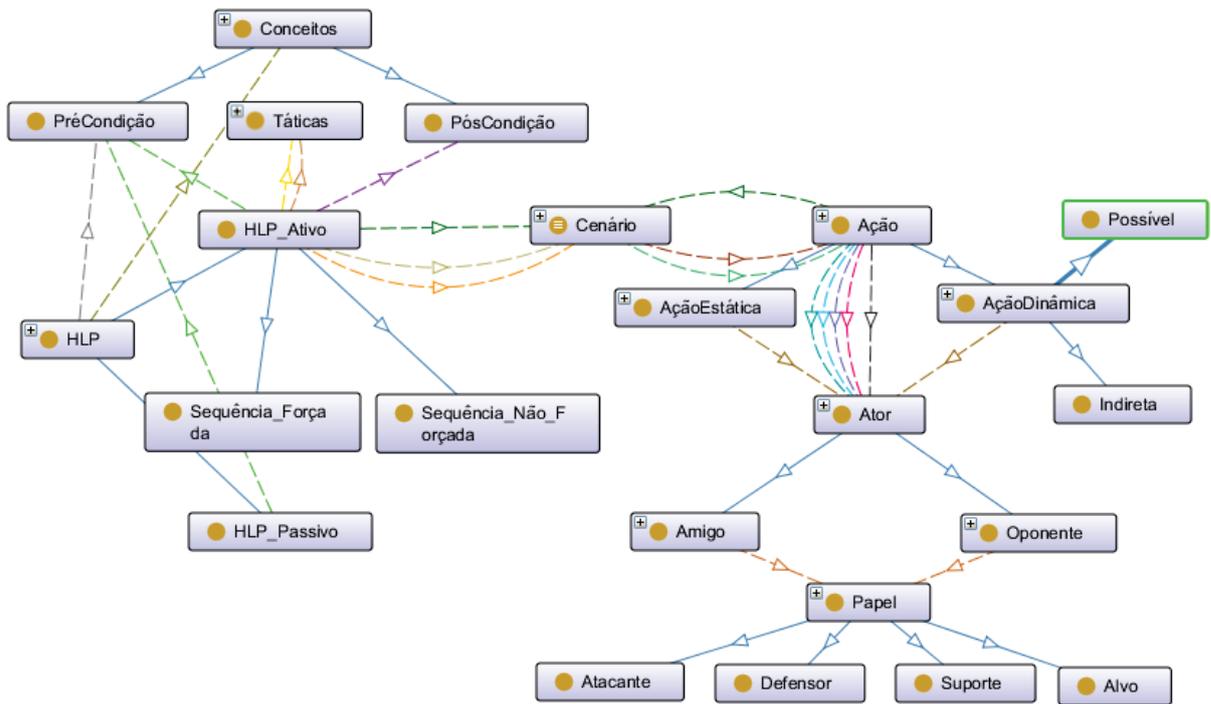


Figura 3.2: Ontologia de HLPs - Relacionamentos

(suporte/coadjuvante) e alvo.

Padrões ativos podem possuir uma prova de conceitos iniciais (PreCondition), seguidos de um plano tático e uma prova de conceitos finais (PostCondition). O plano tático refere-se a sequência de ações que serão realizadas conforme o que está descrito no cenário.

A tabela 3.1 apresenta as propriedades objetos da ontologia e a tabela 3.2 as propriedades de dados pertencentes à classe HLP e seus respectivos tipos. A tabela 3.3 mostra as instâncias das classes: ator, ação estática e ação dinâmica.

Um dos objetivos da definição ontológica sobre HLP é expressar o conhecimento sem ambiguidades, assim as ações dinâmicas foram divididas em duas categorias:

1. Ações indiretas: são aquelas que ocorrem dentro do raio de ação da peça protagonista, fazendo uso de atores intermediários. Para estabelecer o raio de ação de um ator basta considerar para onde o ator pode se mover quando só existe ele no cenário.
2. Ações possíveis: são aquelas que são possíveis de ocorrer através de um movimento tendo como alvo um ator que está fora do raio de ação do ator protagonista.

3.2.2 Linguagem para Representação de HLPs

Com base na taxonomia e definições ontológicas, e como segundo elemento da solução do problema, foi criada uma linguagem de autoria e representação de HLPs (PATLAN). A visualização e identificação correta de HLPs é um fator primordial para a análise estratégica de uma posição, tendo em mente que, pode produzir um ganho

Tabela 3.1: Relação de propriedades objeto da Ontologia

atuaComo temConceitos temAção temAtorPassivo temPósCondição temPréCondição temProtagonista temCenário temTáticas éParteDe éAtorPassivo éProtagonistaDe

Tabela 3.2: Relação de propriedades de dados da classe HLP

Propriedade	Tipo de dado
Atores	["B", "K", "N", "P", "Q", "R", "b", "k", "n", "p", "q", "r"]
ConjuntoExclusivo	["B", "K", "N", "P", "Q", "R", "b", "k", "n", "p", "q", "r"]
NomeDoAutor	string
Descrição	string
PesoHeurístico	float
NomeDoHLP	string

Tabela 3.3: Instâncias por classe

Classe	Instâncias
Ator:Amigo	B,F,K,N,P,Q,R,S
Ator:Oponente	b,f,k,n,p,q,r,s
Ação:AçãoDinâmica	AtaqueIndireto DefesaIndireta AtaquePossível DefesaPossível
Ação:AçãoEstática	Ataca Bloqueia Abaixo Defende AEsquerda Move ANordeste ANoroeste ADireita ASudeste ASudoeste Acima

material, temporal ou posicional, ou pelo contrário, uma perda destes elementos. Portanto a linguagem deve dar suporte para especificar estas características.

Para que isto seja possível a linguagem deve ter o poder de expressar *chunks*² e/ou elementos táticos simples, tais como: garfos, ataques à descoberto (uma forma de ataque indireto), raio-x, entre outros. Estes padrões são de caráter estático, entretanto HLPs devem incluir o caráter dinâmico, pois a sua percepção envolve avaliações temporais e vários movimentos "simultaneamente". Além disto, por se tratar de uma linguagem para autoria de padrões, elementos descritivos e de identificação autoral foram colocados na sintaxe.

O modelo proposto por Finkelstein [24] foi adotado como ponto inicial para a criação da linguagem. Dentre as vantagens, destacam-se: (a) representação tanto de padrões estáticos quanto dinâmicos, (b) possibilidade de tradução semântica para linguagem natural e (c) baseado em grafos direcionados, onde cada vértice representa um ator e as arestas, as relações entre os mesmos. Entretanto, como foi dito na revisão de literatura, o modelo apresentado em [24] é bastante compacto e pode gerar ambiguidades. Devido à isto, a linguagem PATLAN estende os símbolos, as relações e o formato de representação.

Os elementos de tratamento condicionais e verificação de conceitos sobre o padrão foram baseados nas tabelas de conselho encontradas em [8] e os elementos de programação tática foram inspirados no trabalho de Tadepalli [78]. O capítulo 4 contém uma descrição detalhada da linguagem.

3.2.3 Ferramentas de Apoio

Como terceiro elemento de solução, foi definido um conjunto de ferramentas que permitem: autoria de padrões, avaliação sintática, interpretação, reconhecimento com base no conhecimento representado, busca em posições e bases de jogos para testar o formalismo, visualização dos padrões com o uso de MRE, aqui denominado de *StratVision - Um Ambiente para Formalização da Visão Estratégica*.

A figura 3.3 apresenta o arcabouço dos componentes construídos. No capítulo 4 na seção 4.2 encontram-se mais detalhes do funcionamento do StratVision.

3.2.4 Experimentos

A fim de validar o StratVision, a capacidade de representação da linguagem PATLAN e o método de reconhecimento de HLPs, três níveis de experimentos foram realizados:

1. Nível 1: com apoio de enxadristas, todos os casos propostos por Linhares [40] foram codificados. Testes foram executados em todas as posições para cada cenário estratégico usando a ferramenta de autoria.
2. Nível 2: foram escolhidos os padrões de mate Philidor e Anastasia para serem representados na linguagem PATLAN. Buscas foram executadas em jogos onde tais padrões de fato ocorreram. As partidas foram retiradas do sítio especializado www.chessgames.com.

²*Chunk* é uma expressão que foi definida por Miller [47] como sendo a capacidade de armazenar pedaços de informação na memória de curto prazo. Esta capacidade em média tem 7 elementos.

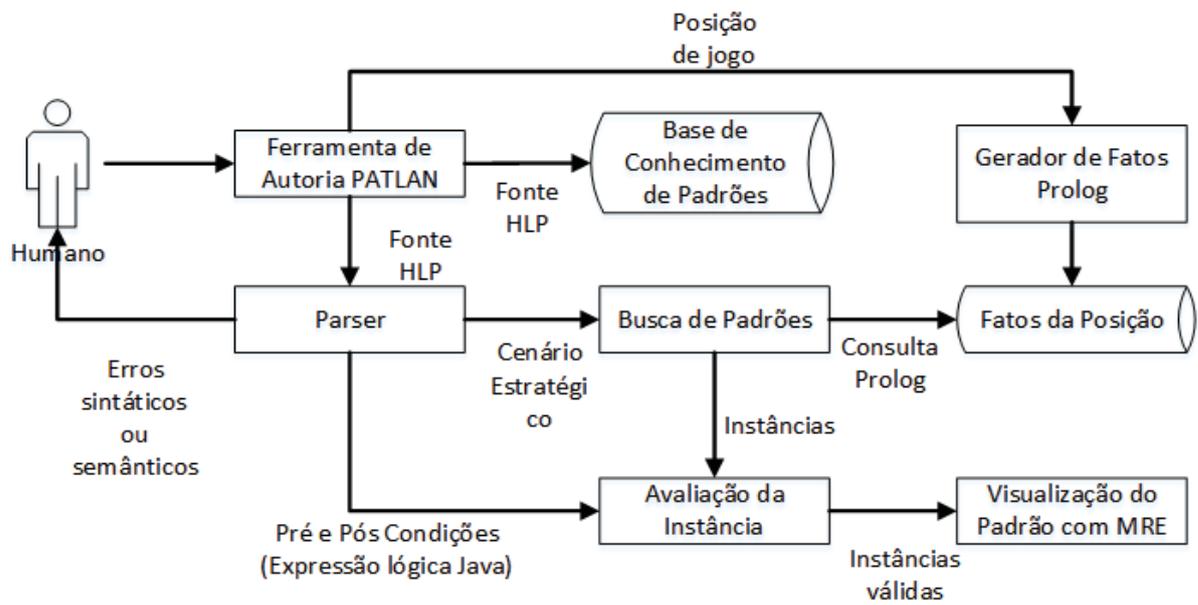


Figura 3.3: Arcabouço do *StratVision*

- Nível 3: buscas em uma base de jogos entre grandes mestres foram realizadas, com mais de 8 milhões de posições analisadas. Todos os 10 casos propostos por Linhares [40] foram alvos das buscas.

Os detalhes dos experimentos e resultados estão no capítulo 5.

4 StratVision - Um Ambiente para Formalização da Visão Estratégica

Este capítulo descreve os elementos que compõe o *ambiente* StratVision. A seção 4.1 descreve a linguagem PATLAN com exemplos de representações. A seção 4.2 descreve os componentes de software desenvolvidos para a representação do conhecimento enxadrístico em nível de visão estratégica.

4.1 PATLAN

A linguagem PATLAN foi criada a partir das definições ontológicas vistas na seção 3.2.1 em três categorias de cláusulas, as quais podem ser escritas em português (sem acentuação¹) ou inglês: (1) identificação/documentação, (2) contexto e (3) especificação.

A identificação é composta das seguintes cláusulas:

- Cláusula PADRAO (PATTERN): deve ser seguida de um texto entre aspas para identificação do padrão sendo codificado.
- Cláusula AUTOR (AUTHOR): deve ser seguida de um texto entre aspas com o nome de quem codificou o padrão.
- Cláusula DESCRICAO (DESCRIPTION): deve ser seguida de um texto entre aspas sobre o padrão.

As cláusulas de contextualização tem por objetivo explicitar quem são os elementos participantes do HLP, e são:

- Cláusula ATORES (ACTORS): declaração explícita de atores (peças) participantes do cenário. Não é obrigatória, porém é útil para declarar atores necessários mas que não participam diretamente das ações, a fim de garantir um mínimo de material na posição.
- Cláusula CONJUNTODEPECAS (SETOFPIECES): cláusula não obrigatória. Por ela o autor pode definir um conjunto exclusivo de peças presentes no tabuleiros. É apropriada para descrever padrões de final de jogo, como por exemplo, finais Torre e Rei, Bispo, Peões e Rei, entre outras.

¹Esta restrição é necessária para evitar conflitos entre tabelas de caracteres da máquina.

As cláusulas de especificação a seguir permitem definir a valoração heurística do HLP, o conjunto de ações que devem estar presentes na posição, os elementos táticos e as condições/conceitos de validação:

- Cláusula PESO (WEIGHT): valor heurístico de caráter subjetivo dado por quem está codificando o padrão. Esse valor pode ser usado no cálculo do movimento plausível como fator de multiplicação sobre o valor heurístico de uma posição, previamente calculado pela FAE, durante uma competição de heurísticas simulada [22]. Valores negativos significam padrões indesejáveis.
- Cláusula CENARIO (SCENARIO): é onde os atores e papeis são definidos propriamente, usando um modelo similar a linguagem DOT para representação de grafos (veja [53] para maiores informações). Os símbolos F e S, representam instâncias de qualquer peça e casa em notação algébrica, respectivamente. Cada linha representa uma ação cujo protagonista (ator ativo) deve estar ao lado esquerdo do tipo da ação. Os atores passivos ficam ao lado direito. As ações são conjuntivas entre si. É possível, porém, criar grupos de ações disjuntivas colocando-as entre {}.
- Cláusula PRECONDICAO (PRECONDITION): cláusula opcional, usada para explicitar características diversas que devem estar presentes para que o padrão seja adequado. Restringe os resultados obtidos do processo de casamento de padrões, por isso deve ser usada com critério, porém em casos como os padrões de mate, evitam falsos positivos.
- Cláusula TATICAS (TACTICALS): cláusula opcional para especificar as ações táticas pertinentes ao padrão. Cada linha representa um movimento que os atores devem executar em sequência.
- Cláusula POSCONDICAO (POSTCONDITION): cláusula opcional que especifica condições que devem estar presentes ao executar todos os movimentos táticos previstos, a fim de garantir que o padrão esteja correto. Essa cláusula só pode ser definida quando a cláusula TATICAS estiver definida.

Por razões de internacionalização e questões técnicas na especificação dos símbolos na gramática da linguagem, justifica-se a adoção da notação enxadrística internacional. Portanto a linguagem PATLAN adota os símbolos especificados na tabela 4.1 para representação de peças.

Uma pequena variação da notação enxadrística é necessária para eliminar ambiguidades na representação. Fica convencionado que os símbolos em caixa alta representam peças do jogador que está analisando a posição e portanto fará o próximo movimento. Os símbolos em caixa baixa representam peças do oponente. Na notação do Xadrez caixa alta refere-se às peças brancas e caixa baixa representa às peças pretas. O coringa "S" independe da caixa pois especifica uma casa ocupada ou vazia para a qual exista um movimento válido, exceto, quando estiver explícito que a casa esteja defendida ou indefesa.

Todo símbolo referente a peças ou casas deve ser seguido de um enumerador sequencial a fim de eliminar ambiguidades referentes as peças duplicadas do tabuleiro. Nas operações de ataque e defesa, pode-se usar uma lista de peças entre parênteses para representar operações dinâmicas, onde o último elemento da lista é o alvo prioritário. Cabe, aqui uma observação: embora o *parser* da linguagem aceite uma lista com mais de dois

Símbolo	Definição
F	Coringa para peça amiga
S	Coringa para casa vazia/ocupada
K	Rei amigo
Q	Dama amiga
R	Torre amiga
N	Cavalo amigo
B	Bispo amigo
P	Peão amigo
f	Coringa para peça oponente
s	Coringa para casa vazia/ocupada
k	Rei oponente
q	Dama oponente
r	Torre oponente
n	Cavalo oponente
b	Bispo oponente
p	Peão oponente
<F>	Peça amiga defendida
<f>	Peça oponente defendida
<S>	Casa defendida por outra peça amiga
<s>	Casa defendida pelo oponente
>F<	Peça amiga indefesa
>f<	Peça oponente indefesa
>S<	Casa não defendida por peças amigas
>s<	Casa não defendida pelo oponente

Tabela 4.1: Notação PATLAN

atores, os algoritmos de interpretação, por razões de simplificação, aceitam apenas dois.

As listagens a seguir contém alguns exemplos de como representar temas táticos (padrões simples) e HLPs. São considerados padrões simples aqueles de caráter estático com até 3 ações e que não expressam táticas e condições finais.

A listagem 4.1 contém o código PATLAN para definir um ataque duplo estático. Um ataque duplo estático ocorre quando uma peça está atacando duas peças simultaneamente (fig. 4.1. A cláusula SCENARIO indica tais ações.

```

1  PATTERN "DOUBLEATTACK"
2  AUTHOR "Luis Bueno"
3  DESCRIPTION "<F1> attacks <f1> and <f2>"
4  WEIGHT 0.8
5  SCENARIO
6    F1 ATTACKS f1
7    F1 ATTACKS f2;
```

Listagem 4.1: Exemplo de representação simbólica de ataque duplo estático

A listagem 4.2 contém o código para definir um ataque duplo dinâmico. Um ataque duplo é dinâmico quando existe a possibilidade de uma peça atacar duas peças simultaneamente. Na figura 4.2 o cavalo (F1) pode se mover para a casa S1 e atacar simultaneamente a Dama e a Torre (f1,f2). A cláusula SCENARIO representa esta ação.

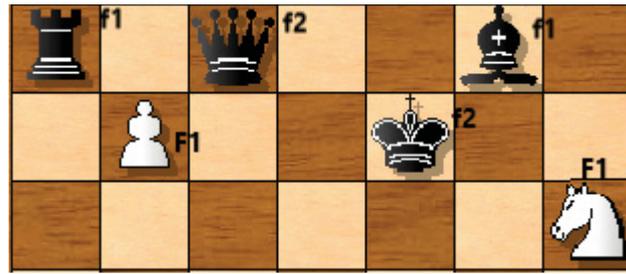


Figura 4.1: Exemplos de instâncias de ataque duplo estático

```

1  PATTERN "DYNAMICDOUBLEATTACK"
2  AUTHOR "Luis Bueno"
3  DESCRIPTION "<F1> moves to <S1> and attacks <f1> and <f2>"
4  WEIGHT 0.8
5  SCENARIO
6    F1 POSSIBLEATTACK (S1,f1)
7    F1 POSSIBLEATTACK (S1,f2);

```

Listagem 4.2: Exemplo de representação simbólica de ataque duplo dinâmico

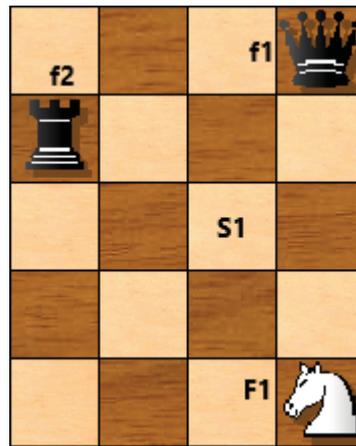


Figura 4.2: Exemplo de instância de ataque duplo dinâmico

Um ataque a descoberto está descrito na listagem 4.3. O cenário indica que existe uma peça F1 que ataca indiretamente através de uma peça amiga F2 a peça oponente f1. Também é necessário que a peça F2 consiga se mover para uma casa S2 qualquer para descobrir o ataque. A figura 4.3 mostra um exemplo de instância deste padrão. A cláusula TACTICALS define como o padrão deve ser executado, ou seja, pelo deslocamento de F2 para S1.

```

1  PATTERN "DISCOVEREDATTACK"
2  AUTHOR "Luis Bueno"
3  DESCRIPTION "<F1> indirectly attacks <f1> through <F2>"
4  WEIGHT 0.6
5  SCENARIO
6    F1 INDIRECTATTACK (F2,f1)
7    F2 MOVE S1
8  TACTICALS
9    F2,S1;

```

Listagem 4.3: Exemplos de representação simbólica de padrões - baixo nível

O tema tático conhecido como ataque raio-x encontra-se definido simplificada e na listagem 4.4 e figura 4.4. Este tema ocorre quando existe uma peça F1 que ataca indiretamente uma peça f2 através de uma peça do oponente f1 que vale mais que f2. Para este ataque se concretizar a peça f1 deve ter a possibilidade de se deslocar para uma outra

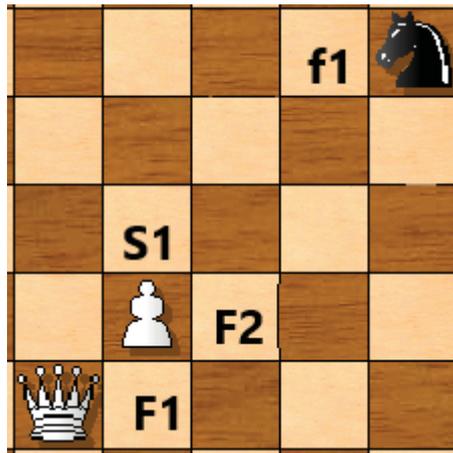


Figura 4.3: Exemplos de instância de ataque a descoberto

casa S1, conforme explícito na cláusula SCENARIO. Outra exigência é que peça F1 deve ser Dama, Torre ou Bispo, conforme declarado na cláusula PRECONDITION.

```

1  PATTERN "X-RAY"
2  AUTHOR "Luis Bueno"
3  DESCRIPTION "<F1> indirectly attacks <f2> through <f1>"
4  WEIGHT 0.6
5  SCENARIO
6    F1 INDIRECTATTACK (f1,f2)
7    f1 MOVES S1
8  PRECONDITION
9    (TYPEOF(F1,Q) OU
10   TYPEOF(F1,R) OU
11   TYPEOF(F1,B)) E
12   VALUEOF(f2) < VALUEOF(f1);

```

Listagem 4.4: Exemplos de representação simbólica de padrões - baixo nível

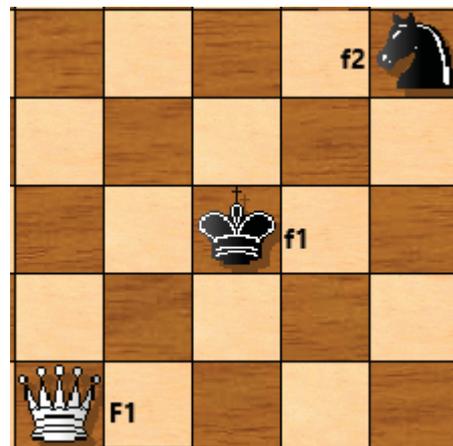


Figura 4.4: Exemplos de instância de ataque raio-x

Supondo que o valor heurístico associado à peões seja 1.0, as regras definidas para o ataque "X-RAY" excluem dos resultados encontrados pelo casador de padrões, pois na realidade um peão pode se mover descobrindo o ataque, mas não pode ser o agente principal do ataque. Os peões podem casar apenas com o símbolo f2, mas não com o símbolo F1 e nem com o símbolo f1 por causa do seu valor. As regras também servem para especificar relações geométricas entre as peças, como distância e posicionamentos específicos, usando funções predefinidas. Um lista destas funções encontram-se no apêndice B. O uso de regras em alguns casos é inevitável, mas o custo de processamento é caro, o que diminui o desempenho.

Pelas definições da ontologia (tabela 3.3), as instâncias dos conceitos foram expressos na forma de funções predefinidas na linguagem PATLAN. A tabela 4.2 apresenta alguns exemplos e sua utilidade. O apêndice B contém um tutorial da linguagem com todas as funções pré-definidas.

Função	Definição
<i>VALORDE</i> (< ator >)	Valor heurístico da peça
<i>CORCASA</i> (< ator >)	Cor da casa onde a peça está
<i>NABORDA</i> (< ator >)	Verdadeiro se o ator está em alguma borda
<i>NOCANTO</i> (< ator >)	Verdadeiro se o ator está em algum canto
<i>COLUNADE</i> (< ator >)	Número da fileira onde está o ator
<i>LINHADE</i> (< ator >)	Número da linha onde está o ator
<i>TIPODE</i> (< ator >, < tipopeça >)	Verdadeiro se o ator é do tipo informado

Tabela 4.2: Exemplos de funções pré-definidas da linguagem PATLAN

Através do modelo de representação, é possível criar bases de conhecimento sobre padrões para compor estudos de caso. Representar estes padrões pela linguagem DHJOG (sucintamente apresentada no capítulo introdutório) nos moldes atuais seria bastante trabalhoso e em certos casos impossível de fazê-lo explicitamente, como por exemplo o "AtaqueDuploDinâmico". É importante destacar que isto não denota uma fraqueza da DHJOG, mas apenas o fato de que foi projetada para avaliações estáticas em uma posição.

Vale neste ponto dizer que o padrão deve sempre ser formalizado a partir de uma posição em que o próximo movimento pertence ao lado que fará a busca. Também é importante que todos os elementos atuantes em um padrão estejam explícitos.

Existe uma cláusula reservada para definir o peso do padrão. A ideia é que futuramente possa se atribuir um valor de relevância estatístico, que permita classificar se este padrão é forte ou fraco, quando houver incerteza sobre sua utilidade. Uma forma de fazer isto seria, uma vez definido um padrão, rodar o algoritmo de reconhecimento em uma extensa base de jogos já realizados para identificar sua frequência e qual foi o resultado das partidas onde eles ocorreram, fazendo uma ponderação dos resultados de sorte que quanto mais vitórias mais alto seja o seu peso. Padrões de mate devem receber um valor alto pela sua própria natureza, independente dos resultados de uma busca.

A linguagem PATLAN apresenta um diferencial que é o fato de ser bilíngue, dando suporte para padrões descritos tanto em português quanto em inglês. A listagem 4.5 mostra um garfo dinâmico descrito com as cláusulas em inglês.

```

1 PATTERN "DYNAMICFORK"
2 AUTHOR "Luis Bueno"
3 DESCRIPTION "<P1> can attack <f1> and <f2> through <S2>"
4 WEIGHT 0.8
5 SCENARIO
6   P1 POSSIBLEATTACK (S1,f1)
7   P1 POSSIBLEATTACK (S1,f2);

```

Listagem 4.5: Exemplo de representação simbólica em língua inglesa

Outro diferencial é a forma como o Parser foi construído que facilita o processo de tradução do HLP representado em linguagem natural, possibilitando o uso de múltiplas representações no aprendizado de padrões. O StratVision já possui suporte parcial para isto, apesar de não estar no propósito da presente tese.

4.2 Componentes de Software do Ambiente StratVision

O *Ambiente StratVision* foi elaborado para permitir humanos explicitarem seu conhecimento sobre padrões. A figura 3.3 mostra o arcabouço principal composto das seguintes partes: ferramenta de autoria, base de fatos, gerador de fatos, casamento de padrões, avaliador das instâncias e visualização dos fatos. Foi integrado à ferramenta Heuchess+ [11], por razões de reaproveitamento de estruturas de dados, classes e métodos úteis para o StratVision. O arcabouço descreve como o fluxo de dados ocorre no ambiente e ocorre da seguinte maneira:

1. Humano (aprendiz/especialista) acessa a ferramenta de autoria onde seleciona uma posição de jogo na qual exista o padrão que pretende representar.
2. Humano descreve o padrão seguindo as regras e convenções da linguagem PATLAN.
3. O código fonte é salvo na base de conhecimento de padrões.
4. Ao executar a busca os seguintes passos são realizados:
 - (a) A posição de jogo é enviada para o gerador de fatos Prolog que gera um arquivo de memória com os fatos encontrados na posição. O apêndice C contém um exemplo de fatos gerados de acordo com as regras de predicados do Prolog.
 - (b) O código fonte HLP é enviado para o Parser que o avalia sintática e semanticamente e gera as tabelas de símbolos para interpretação do código fonte. O código é desmembrado em 4 partes: cenário, pré-condições, táticas e pós-condições.
 - (c) O cenário é traduzido para uma consulta Prolog e enviado para o módulo de busca que aciona o TuProlog sobre a base de fatos da posição. Este retorna uma lista de instâncias encontradas que combinam com o cenário.
 - (d) Na sequência, para cada instância da lista são executadas a avaliação das pré-condições, das táticas e pós-condições, quando for o caso. Uma lista de instâncias válidas é gerada.
5. As instâncias são visualizadas na própria ferramenta de autoria.

A seção 4.2.1 descreve as características da ferramenta de autoria. A seção 4.2.2 descreve o módulo gerador da base de fatos. A seção 4.2.3 apresenta o módulo de avaliação sintática e semântica da linguagem PATLAN. A seção 4.2.4 descreve como é realizado o processo de busca e casamento dos padrões. Finalmente, a seção 4.2.5 descreve como são aceitas/rejeitadas as instâncias do cenário encontradas na posição de jogo em análise.

4.2.1 Ferramenta de autoria

A figura 4.5 mostra a interface da ferramenta de autoria. A interface faz uso das múltiplas representações externas (MRE) da seguinte forma: a posição é visualizada graficamente e no formato FEN, as instâncias encontradas são visualizadas em modo texto e graficamente destacadas na posição. Na figura está exemplificado o código do padrão "Ataque duplo dinâmico" onde uma busca no modo BRANCAS (*WHITE*) foi executada sobre a posição carregada no tabuleiro. O resultado do reconhecimento do

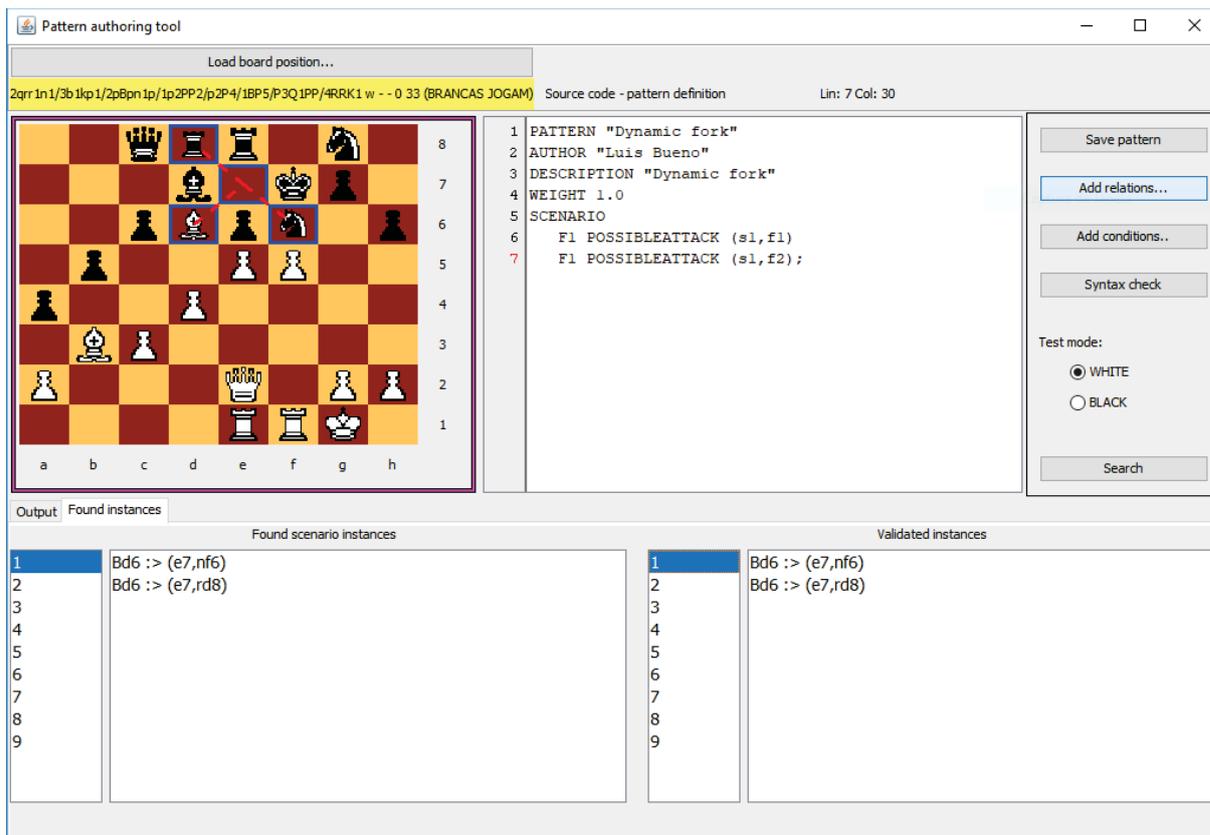


Figura 4.5: Ferramenta de autoria PATLAN - *StratVision*

padrão é mostrado em duas listas, na parte inferior. A lista à esquerda mostra as instâncias brutas, que combinam com o cenário descrito no código fonte. A lista à direita, mostra as instâncias que passaram nas pre-condições, na execução das táticas e nas pós-condições, quando definidas. No exemplo da figura, como nenhum destes elementos foi definido, as duas listas são idênticas.

A interface mostra cada componente do cenário em notação algébrica e faz as ligações visuais no tabuleiro. Linhas vermelhas tracejadas significam ataques indiretos. Linhas vermelhas sólidas significam ataques diretos. Linhas verdes tracejadas significam defesas indiretas. Linhas verdes sólidas significam defesas diretas.

4.2.2 Gerador de Fatos

O módulo gerador de fatos é responsável por executar uma varredura na posição foco da busca, e gerar uma base de fatos com predicados Prolog especialmente definidos para permitir o casamento de padrões a partir do cenário especificado. O *StratVision* armazena uma posição de jogo internamente na forma de *bitboards*.

Segundo Slate e Atkin [66], o grupo KAISSA da antiga União Soviética inventou a técnica de *bitboards* na década de 1960. Um *bitboard* é uma estrutura de dados na forma de mapa de *bits*² que permite armazenar as informações de uma posição de jogo como xadrez. No caso do tabuleiro de xadrez são bastante eficazes para geração de movimentos e diversas

²Um mapa de *bits* é uma sequência de *bits* de tamanho finito que mapeiam a informação de um domínio específico. Cada bit pode valer 0 ou 1.

inferências estáticas, uma vez que utilizam operadores binários e máscaras numéricas para realizar estas operações. Com advento das máquinas de 64 bits se tornaram ainda mais eficientes, pois uma única palavra consegue armazenar informação das 64 casas do tabuleiro.

Entretanto, são necessários 12 mapas de bits para representar uma posição. Para cada tipo de peça é montado um mapa, onde o valor 1 indica que a casa correspondente no tabuleiro está ocupada por aquele tipo de peça. A partir destes 12 mapas é possível rapidamente calcular quais casas estão ocupadas, quais vazias, quais estão sendo atacadas, quais defendidas, para onde as peças podem se mover, etc.

No apêndice C encontra-se um exemplo de base de fatos gerada em função da varredura da posição exemplificada na figura 4.5.

4.2.3 *Parser*

O módulo Parser é responsável pela avaliação sintática e semântica do código fonte PATLAN. Além disto, ele faz a interpretação e tradução do código para o módulo de reconhecimento. Isto é realizado da seguinte forma:

- O conteúdo das cláusulas ATORES e CONJUNTODEPECAS é armazenado em estruturas internas especiais para uso no módulo de reconhecimento.
- O conteúdo da cláusula CENÁRIO é traduzido para uma sentença de consulta Prolog.
- O conteúdo das cláusulas PRECONDICAO e o da POSCONDICAO são traduzidos para expressões lógicas Java, através de um mapeamento pré-definido entre as funções PATLAN e funções Java que estão programadas no módulo de reconhecimento.
- O conteúdo da cláusula TATICAS é armazenado numa tabela interna o qual será devidamente interpretado e executado no módulo de avaliação das instâncias.

A figura 4.6 mostra a saída da busca realizada no exemplo da figura 4.5. A consulta Prolog é mostrada na linha "Fato buscado".

Método de Tradução para Prolog

A tradução de um cenário descrito em PATLAN para Prolog é realizado parte pelo Parser e parte pela classe HLP (conforme definido na figura 3.1. O Parser gera uma tabela de símbolos que classifica cada *token* do cenário como sendo: Ator (amigo ou oponente) ou Ação (identificando a ação). Então a classe HLP pro-

Output	Found instances
<p>Tudo certo!</p> <p>Padrão buscado [Dynamic fork]</p> <p>Fato buscado [meu(MF1),seu(Sf1),seu(Sf2),ataquepossivel(MF1,Ss1,Sf1),ataquepossivel(MF1,Ss1,Sf2)]</p> <p>Instâncias encontradas: 9</p>	

Figura 4.6: Saída do processo de busca de padrões - *StratVision*

cessa a tabela de símbolos de acordo com o algoritmo simplificado da listagem 4.6.

```

1 inicio
2  Entrada: Cenario
3  Saida : Prolog
4  Prolog = ""
5  // prepara sentencas gerais da consulta
6  para cada Acao em Cenario
7    para cada Ator em Acao.Atores
8      se Ator e Amigo
9        Ator.Texto = "M" + Ator.Texto
10       Prolog += "meu(Ator.Texto + ","
11
12       se Ator esta Protegido
13         Prolog += "protegidapormim(M" +
14           Ator.Texto + ","
15           "M" + Ator.Texto + ")"
16       fim se
17       senao
18         Ator.Texto = "S" + Ator.Texto
19         Prolog += "seu(" +
20           Ator.Texto +
21           "),"
22       se Ator nao e Coringa
23         Prolog += Ator.Tipo + "(" +
24           Ator.Texto + "),"
25       fim se
26       se Ator esta Protegido
27         Prolog += "protegidaporti(" +
28           Ator.Texto +
29           "),"
30       fim se
31       fim se
32       se Ator nao e Coringa
33         Prolog += Ator.Tipo + "(" +
34           Ator.Texto + "),"
35       fim se
36       fim para
37     fim para
38     // produz as sentencas para cada acao
39     // os atores ja estao corretamente nomeados para
40     // aplicar na consulta prolog
41     // se amigo possui o prefixo "M"
42     // se oponente possui o prefixo "S"
43     para cada Acao em Cenario
44       se Acao e Estatica
45         Prolog += Acao.predicadoProlog +
46           "(" +
47           Acao.Protagonista +
48           "," +
49           Acao.Alvo +
50           "),"
51       senao
52         Prolog += Acao.predicadoProlog +
53           "(" +
54           Acao.Protagonista +
55           "," +
56           Acao.Suporte +
57           "," +
58           Acao.Alvo +
59           "),"
60       fim se
61     fim para
62     retorna Prolog
63 fim

```

Listagem 4.6: Algoritmo de tradução PATLAN para Prolog

Assim o código do cenário da listagem 4.5, é traduzido para a seguinte expressão em PROLOG: $meu(MP1)$, $p(MP1)$, $seu(Sf1)$, $seu(Sf2)$, $ataquepossivel(MP1,MS1,Sf1)$, $ataquepossivel(MP1,MS1,Sf2)$.

O apêndice C contém a relação dos predicados e regras Prolog definidos para identificar se um cenário representado em PATLAN está presente numa posição.

4.2.4 Busca de padrões - módulo de reconhecimento

O módulo de reconhecimento utiliza os parâmetros interpretativos gerados pelo *Parser* para realizar sua tarefa, adotando os seguintes passos:

1. Gerar *bitboards*: os *bitboards* da posição em formato FEN são gerados.
2. Verificação de minimalidade material da posição: antes de prosseguir, o algoritmo verifica se a posição atende os pré-requisitos materiais mínimos constantes nas cláusulas ATORES, CONJUNTODEPECAS E CENARIO. Caso contrário encerra-se a busca retornando zero instâncias brutas.
3. Geração da base de fatos: o módulo gerador de fatos Prolog é acionado. A base de fatos é enviada para o módulo TuProlog.
4. Consulta Prolog: a consulta Prolog traduzida pelo interpretador é passada para o TuProlog que devolve as instâncias encontradas na posição, quando houverem.
5. Estruturação MRE: uma rotina é acionada para gerar estruturas de dados internas que permitirão a visualização com MRE.

As instâncias devolvidas pelo TuProlog são denominadas de instâncias brutas e serão passadas para o módulo de avaliação, descrito a seguir.

4.2.5 Avaliação das instâncias

Uma vez encontradas instâncias brutas e havendo pré-condições definidas no código fonte, o procedimento de validação de conceitos é ativado recebendo como parâmetros de entrada: a pré-condição, as táticas, as pós-condições, as instâncias brutas e o *bitboard* e retorna a lista de instâncias válidas. A listagem 4.7 contém o pseudo-algoritmo em Java para avaliação de instâncias. Caso não hajam pré-condições ou pós-condições as instâncias brutas são adicionadas à lista de instâncias válidas.

```

1 ArrayList<String> avaliaInstancias(String precondicao,
2   String poscondicao, Taticas taticas,
3   ArrayList<String> instbrutas, Bitboard bit)
4 {
5   // classe para execucao do codigo Java
6   ScriptEngine interpret = new ScriptEngine();
7   boolean resultOK;
8   ArrayList<String> instanciasOK = new ArrayList<>();
9
10  // para cada instancia bruta encontrada na posicao
11  for(String instancia : instbrutas)
12  {
13    if(poscondicao.isEmpty()) // se nao existem precondicoes
14    {
15      resultOK = true;
16    }
17    else
18    {
19      resultOK = interpret.eval(precondicao,instancia, bit);
20    }
21    if(resultOK)
22    {
23      if(poscondicao.isEmpty()) { // se nao existem poscondicoes
24        resultOK = true;
25      } else {

```

```
26     bit .executarTaticas(taticas); // taticas
27     resultOK = interpret.eval(poscondicao,instancia,bit);
28     bit .desfazerTaticas(taticas);
29 }
30 if(resultOK)
31 {
32     instanciasOK.add(instancia)
33 }
34 }
35 }
36 return instanciasOK;
37 }
```

Listagem 4.7: Base de fatos Prolog

4.3 Resumo

Este capítulo apresentou os elementos do ambiente StratVision, sendo: linguagem de formalismo de HLPs (PATLAN) e os componentes de software. O ambiente é composto dos seguintes módulos: (a) ferramenta de autoria, (b) parser, (c) gerador de fatos, (d) busca de padrões, (e) avaliador de instâncias e (f) visualizador de instâncias, incorporado à ferramenta de autoria.

5 Experimentos

Este capítulo apresenta os experimentos de representação e busca de HLPs realizados para comprovação das hipóteses formuladas na seção 1.5. Os experimentos tiveram como ponto de partida as posições apresentadas por Linhares [40], para as quais foram criadas representações e testes preliminares para cada par de posição. Em todas as posições o StratVision encontrou instâncias dos HLPs representados.

De acordo com a ontologia definida na seção 3.2.1 existem duas classes de HLPs: (1) HLPs ativos e (2) HLPs passivos.

HLPs ativos são aqueles que dependem de um cenário no qual um conjunto de ações estejam em andamento. Se subdividem em duas categorias:

1. Sequencia forçada de lances: uma vez que o padrão surge e as táticas corretas sejam aplicadas, o oponente não terá o que fazer para evitar que o HLP se concretize.
2. Sequencia não forçada de lances: neste caso existe, em alguns cenários, a possibilidade do oponente interpor-se com alguma peça impedindo a execução do padrão. Evidentemente, isto vai depender do arranjo de peças no tabuleiro.

HLPs passivos são aqueles que não necessitam da formação de um cenário específico no qual as ações táticas envolvem elementos do cenário, mas apenas exigem prova de conceitos e nos quais as ações táticas estão relacionadas a estes conceitos porém são de longo prazo e é custoso definir quantos lances a frente o padrão se concretizará por completo, o que pode ser visto como algo puramente estratégico.

Dentre as posições de Linhares [40], as posições das figuras 5.2, 5.3, 5.15 foram escolhidas para experimentos mais aprofundados, pois representam adequadamente as duas classes ontológicas de HLPs.

Portanto, o presente capítulo encontra-se assim dividido: a primeira parte apresenta os estudos com HLPs ativos (seção 5.1), particularmente os de xeque-mate. A segunda parte descreve os experimentos de busca com HLPs passivos (seção 5.2).

5.1 Representação de HLPs ativos

Neste trabalho vários HLPs ativos foram codificados a fim de testar a capacidade de representação da linguagem PATLAN e os métodos de reconhecimento, tais como os padrões de mate: Philidor, Anastasia, Cozio, Bonde, entre outros. Apenas Philidor e Anastasia encontram-se descritos neste texto nas seções 5.1.1 e 5.1.2. O Philidor é um típico exemplo de HLP ativo com sequência forçada de lances, e o Anastasia é um típico exemplo de HLP ativo com sequência não forçada de lances. O Philidor necessita de 3 *plies* para ser executado e o Anastasia apenas um. Para estes dois foi dedicado mais tempo em experimentos e no ajuste do conhecimento representado a fim de aumentar a acurácia e diminuir os casos de falsos positivos.

5.1.1 HLP com sequência forçada de lances - Xeque-mate Philidor

De acordo com o sítio especializado:

http://www.chessgames.com/perl/chess_collection?cid=1001808

(acesso em 02/11/2018), o primeiro registro deste padrão ocorreu em 1497, numa publicação de Luis Ramirez de Lucena. Foi usado por Linhares [40] em seus experimentos sobre visão estratégica. É conhecido também como mate sufocado e designa uma posição onde o rei de um dos lados leva mate de um Cavalo pois está imobilizado por suas próprias peças. Costuma envolver um sacrifício, em geral de Dama, porém a ideia central é atrair uma peça adversária para uma casa que deixe o rei sufocado. Outra característica observada em todas as posições de convergência estudadas é que o rei adversário já está sufocado.

A figura 5.1 mostra um exemplo da execução das sequências forçadas do Philidor. A primeira posição é o ponto onde o padrão surge e indica o primeiro lance. É importante observar o posicionamento do cavalo que tanto protege a casa b8 para onde a Dama irá se deslocar aplicando xeque, quanto se deslocando para a casa c7 que resulta em xeque. Na segunda posição, a Torre adversária deverá se deslocar para a casa c7, capturando a Dama e deixando o rei sufocado. A terceira posição mostra o lance em que o Cavalo fará a execução final do padrão aplicando xeque-mate no oponente, como demonstrado na última posição.

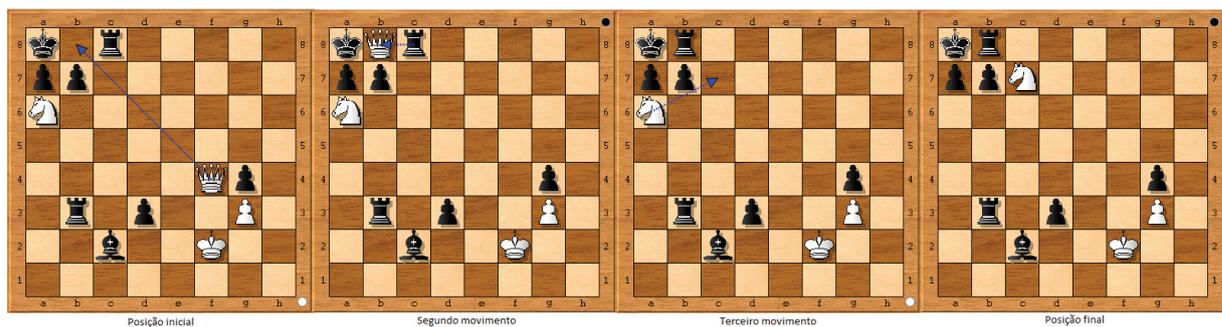


Figura 5.1: Exemplo de execução do Philidor - Fonte: adaptado de [40]

A listagem 5.2 mostra a representação em PATLAN. Esta representação obteve 100% de acurácia nos experimentos das seções 5.1.1 e 5.1.1 em todas as partidas que atingiram a posição de convergência do padrão e baixo índice de falsos positivos no experimento 5.1.1. Como pode ser observado nas listagens 5.1 e 5.2, pensar sobre os elementos que compõe um padrão não é uma tarefa trivial, deve ser realizado de forma incremental e, preferencialmente testado em um conjunto de posições bem variadas. A listagem 5.1 mostra a primeira versão deste código, que foi eficaz para as posições das figuras 5.2 e 5.3, mas ineficaz para posições onde o padrão ocorre em locais diferentes do tabuleiro.

```

1 PATTERN "PHILIDOR 1.0"
2 AUTHOR "Luis Bueno"
3 DESCRIPTION "Philidor 01"
4 WEIGHT 999.0
5 SCENARIO
6   N1 POSSIBLEATTACK (S1,k1)
7   F1 POSSIBLEATTACK (S2,k1)
8   r1 DEFENDS S2
9   k1 DEFENDS p1
10 PRECONDITION
11  RANKOF(k1) = RANKOF(r1) AND

```

```

12 (FILEOF(p1) = 8 OR FILEOF(p1) = 1) AND
13 (RANKOF(k1) = 8 OR RANKOF(k1) = 1) AND
14 (TYPEOF(F1,Q) OR TYPEOF(F1,R))
15 ;

```

Listagem 5.1: Philidor - primeira versão

```

1 PATTERN "LINHARES_08_20_B"
2 AUTHOR "Author name"
3 DESCRIPTION "Philidor pattern style"
4 WEIGHT 999.0
5 SCENARIO
6 N1 POSSIBLEATTACK (S1,k1)
7 F1 POSSIBLEATTACK (<S2>,k1)
8 f1 DEFENDS S2
9 k1 DEFENDS S2
10 k1 DEFENDS f2
11 f3 NOT INDIRECTDEFENSE (F1,S2)
12 PRECONDITION
13 (DIRECTDEFENSES(S2,OPONENT) = 1 OR
14 (DIRECTDEFENSES(S2,OPONENT) = 2)) AND
15 (DIRECTDEFENSES(S1,OPONENT)=0 OR
16 RELDEFENSESVALUE(S1,OPONENT)=RELATIVEVALUEOF(f1)) AND
17 MOVESOF(k1) = 0 AND
18 (TYPEOF(F1,Q) OR
19 TYPEOF(F1,R)) AND
20 (TYPEOF(f1,R) OR
21 TYPEOF(f1,N))
22 TACTICALS
23 F1,S2
24 f1,S2
25 N1,S1
26 POSTCONDITION
27 MOVESOF(k1) = 0;

```

Listagem 5.2: Philidor - versão experimental inicial

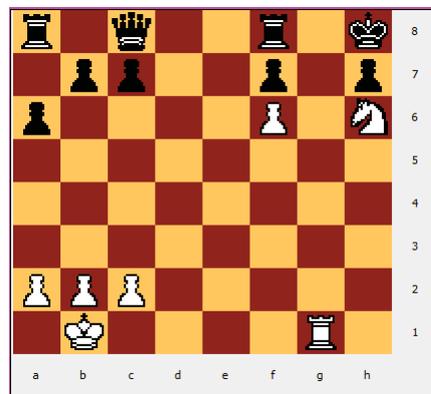


Figura 5.2: Estudo de caso - Fonte: [40]

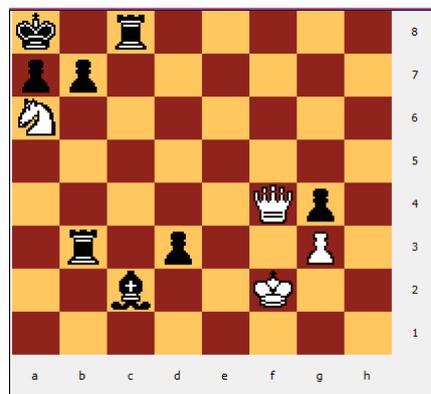


Figura 5.3: Estudo de caso - Fonte: Adaptado de [40]

As linhas 1 a 3 da listagem 5.2 definem a parte documental de autoria do Padrão. A cláusula WEIGHT define o valor heurístico do padrão. Neste caso é um valor alto por ser altamente desejável encontrá-lo em uma partida.

A cláusula SCENARIO descreve as ações que devem estar ocorrendo no tabuleiro, preferencialmente em termos abstratos, isto é, sem estabelecer casas ou regiões específicas para que as ações ocorram. Elenca os atores que participam conforme o que foi explicado na figura 5.1 e as ações que eles devem estar executando. Na linha 6, o cavalo entra cena podendo atacar o rei pela casa S1 (c7). Na linha 7 está explicitado que a máquina deve encontrar qualquer peça amiga que se mova para uma posição S2 (b8) defendida por outra peça amiga e dali ataque o rei adversário. A linha 8 define que deve existir uma peça adversária defendendo a casa S2 (b8). A linha 9 indica que o rei oponente deve estar defendendo a casa S2 (b8) e a linha 10 que ele também deve estar defendendo uma peça próxima a ele. A linha 11 é a exceção, significando que **não** deve existir alguma peça do adversário que consiga defender a posição S2 (b8) através de F1, o que abriria uma possibilidade de contra-golpe descaracterizando o padrão.

Somente especificar um cenário não é suficiente para encontrar o Philidor. Existem outros elementos que devem ser checados ou na forma de conceitos através de funções de apoio predefinidas na PATLAN. A cláusula POSTCONDITION estabelece um conjunto de condições que devem estar presentes na posição de convergência. As linhas 13 e 14 estabelecem que a casa S2 (b8) deve ter uma ou no máximo duas defesas diretas do oponente. Existem casos em que além da Torre, pode existir um cavalo defendendo a posição S2 (b8) e o padrão se concretizar. As linhas 16 e 17 também assertam que o número de defesas diretas do oponente sobre casa S1 (c7) devem ser igual a zero ou que a defesa venha exclusivamente a partir da peça f1 (Torre). As funções DIRECTDEFENSES e RELDEFENSESVALUE não incluem o rei na contagem.

A linha 17 estabelece que o rei oponente esteja sufocado. Isto é necessário, pois em casos como os da figura 5.2, se o peão em f6 não estivesse protegendo a casa g7, o rei conseguiria escapar do xeque-mate.

As linhas 18 a 21 estabelecem quais os tipos específicos de peça para F1 e f1 que podem ser Dama ou Torre, e Torre e Cavalo, respectivamente.

A cláusula TACTICALS deixa pré-programada a sequência de ações táticas para a execução do padrão. Cada linha explicita um movimento, começando pela peça instanciada para F1 (Dama) que se move para S2 (b8), em seguida a peça instanciada para f1 (Torre) se move para S2 (b8). E finaliza com o movimento de N1 (Cavalo) para S1 (c7).

A cláusula POSTCONDITION deixa explícito que a condição final obtida deve ser o sufocamento do rei (linha 27).

Experimento 1

O primeiro experimento, conforme descrito na seção 3.2.4, foi executado sobre as posições das figuras 5.2 e 5.3. Importante ressaltar que a segunda posição foi alterada a fim de atingir a posição de convergência do padrão. As figuras 5.4 e 5.5 mostram o resultado da busca na ferramenta de autoria.

Experimento 2

Neste experimento foi executada uma busca em 40 partidas (documentadas pelo sítio <http://www.chessgames.com/perl/chesscollection?cid=1001808> - acesso em 18/11/2018) entre grandes mestres onde o padrão de mate Philidor surge.

uma vez que é o cálculo da vantagem material que tem maior peso dentro da FAE.

O padrão foi encontrado em 27 posições. Em 13 partidas o oponente desistiu antes do padrão se configurar e de propósito não foram inclusos na tabela.

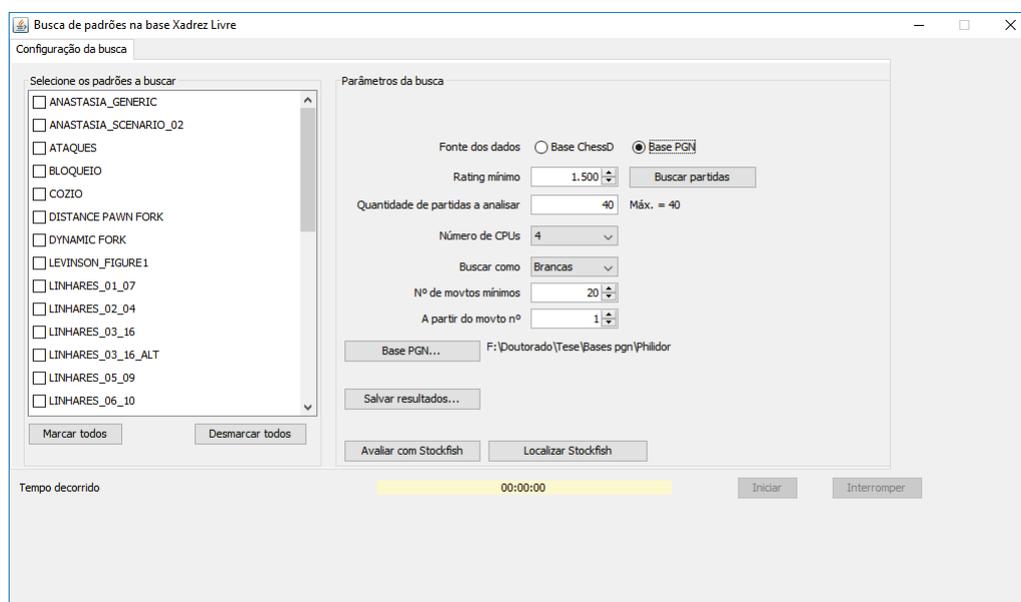


Figura 5.6: Interface para busca em bases pgn

O valor estático retornado pela máquina Stockfish é positivo quando as brancas estão com vantagem e negativo quando as pretas estão e a medida é dada em peões¹. Se, além das posições onde a avaliação foi de vitória para o oponente, considerar-se também as posições onde o valor é entre -1 e 1, que significa uma vantagem menor que 1 peão e as posições avaliadas como derrota pelo lado vencedor, temos que: 67,8% das posições possivelmente seriam evitadas ou excluídas da avaliação pelas podas alfa-beta no cálculo MINIMAX.

Alguns desses jogos apresentaram mais de um cenário que combinavam com o especificado no padrão, porém as condições iniciais garantiram que o StratVision não resultasse como positivos. A tabela 5.2 apresenta as partidas, a posição, o número do *ply* onde ocorreram os cenários similares e o número do *ply* onde de fato ocorreu o padrão. No caso da posição B03, apesar de existir um xeque-mate em 1 lance, este não confere com o padrão Philidor que é uma sequência de 3 plies.

Experimento 3

Este experimento foi realizado no servidor "servheuchess", uma máquina virtual do laboratório C3SL do Departamento de Informática da Universidade Federal do Paraná. O servidor possui 12 núcleos de processamento que foram utilizados para paralelizar as buscas em uma base com 275.606 partidas, todas entre grandes mestres enxadristas com ELO acima de 2.500, disponível no sítio <https://chess-db.com/public/downloads/gamesfordownload.jsp> - acesso em 10/09/2017.

Devido à problemas de perda de conexão constantes com os servidores, por estouro de tempo o que acabava interrompendo o processo, optou-se por fazer uma divisão da

¹Informação técnica disponível em <http://support.stockfishchess.org/discussions/questions/1303-stockfish-evaluation-numbers-and-their-meaning> - acesso 06/11/2018

Tabela 5.1: Resultados busca Philidor - chessgames.com

Jogo	Posição de convergência encontrada pelo StratVision	StockFish
Jean-Jacques Rousseau x David Hume	Figura 5.7 B01	-5,77*
John Cochrane x Bonnerjee Mohishunder	Figura 5.7 B02	-2,69*
Paul Morphy x Thomas Jefferson Bryan	Figura 5.7 B03	-0,19*
Paul Morphy x Schrufer	Figura 5.7 B04	-3,24*
Bird Dobell x NN	Figura 5.7 B05	-2,25*
Franklin Knowles Young x L Dore	Figura 5.7 B06	-7,25*
Siegbert Tarrasch x Kolb	Figura 5.7 B07	7,07
Georges Koltanowski x J Salazar	Figura 5.7 B08	1,4
Friedrich Saemisch x Oskar Menzinger	Figura 5.7 B09	-3,08*
Claude Frizzel Bloodgood x B Evans	Figura 5.7 B10	0,3
Pal Benko x Israel Albert Horowitz	Figura 5.7 B11	0,2
Jan Timman x Nigel Short	Figura 5.7 B12	1,52
Holger Namyslo x Ralf Lau	Figura 5.7 B13	3,78
Alexander Grischuk x Ruslan Ponomariov	Figura 5.7 B14	-2,88*
Klaus Bischoff x Karsten Mueller	Figura 5.7 B15	-1,09*
Helmut Reefsclaeger x Corrado Astengo	Figura 5.7 B16	-2,49*
Betul Cemre Yildiz x Lucija Batory	Figura 5.7 B17	4,02
Medina Warda Aulia x Kieran Lyons	Figura 5.7 B18	4,15
	Média	-0,472
Schaaf x Carl Hartlaub	Figura 5.8 P01	-2,4
James McConnell x Paul Morphy	Figura 5.8 P02	-1,04
Vania F. Da Vilhete x Rodolfo A. Formoso	Figura 5.8 P03	-0,9
NN x Terry McCracken	Figura 5.8 P04	-0,79
James Moore Hanham x Wilhelm Steinitz	Figura 5.8 P05	-5,17
NN x Gioachino Greco	Figura 5.8 P06	0,84*
NN x Gioachino Greco	Figura 5.8 P07	0,92*
Samsonov x Rashid G. Nezhmetdinov	Figura 5.8 P08	1,63*
Sari E. D. Godoy x Veronica G. Castro	Figura 5.8 P09	3,89*
	Média	-0,335

Jogo	Ply negativo	Ply positivo	Posição
Tarrasch x Kolb	41	55	Figura 5.9 B01
Smirin x Ribeiro	29	-	Figura 5.9 B02
Smirin x Ribeiro	31	-	Figura 5.9 B03
Aulia x Lyons	31	65	Figura 5.9 B04

Tabela 5.2: Cenários similares ao Philidor encontrados

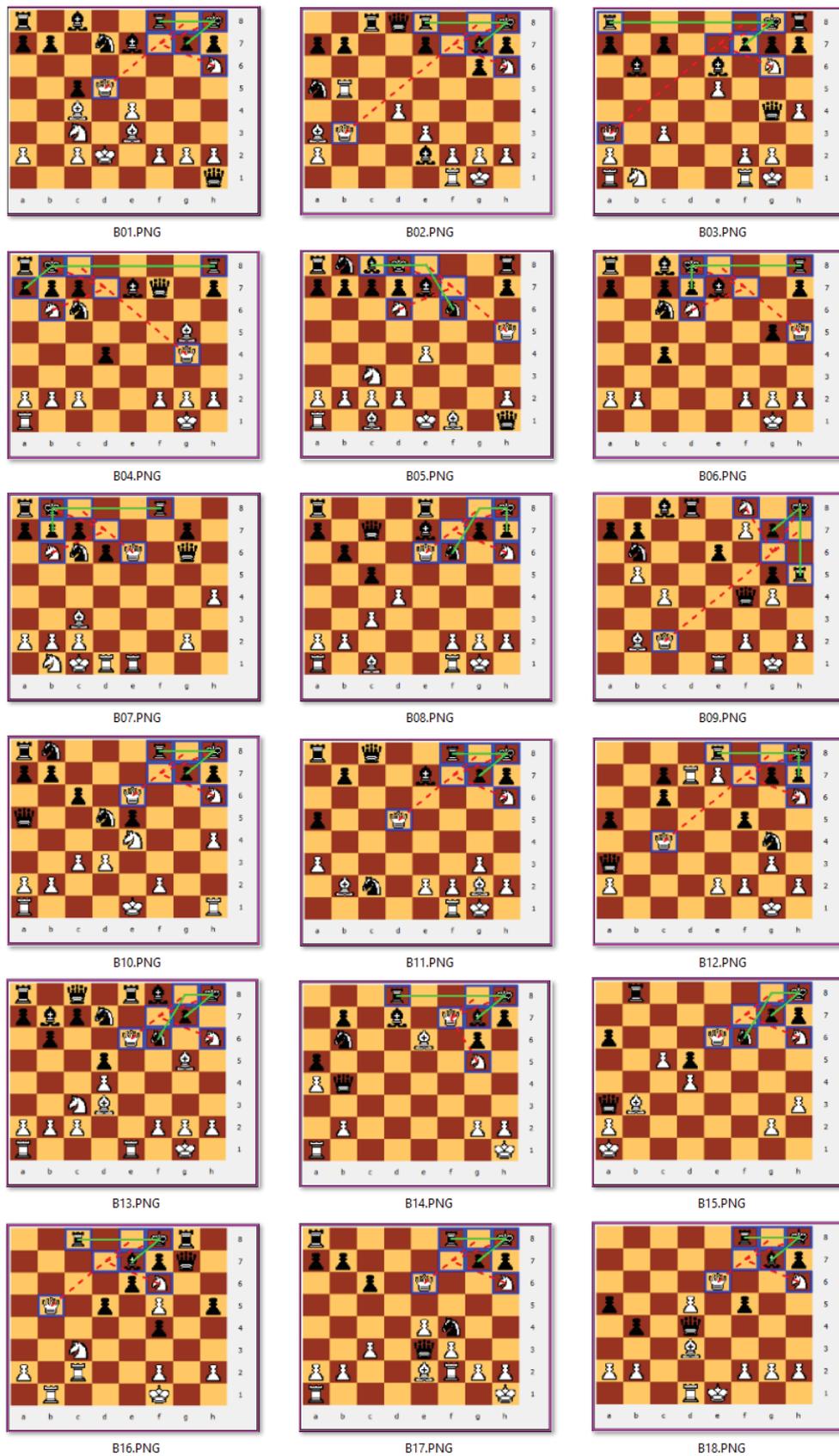


Figura 5.7: Posições de convergência encontradas pelo StratVision - Brancas - Experimento 2

amostragem em 8 lotes de aproximadamente 14.740 partidas totalizando 117.910 partidas e



Figura 5.8: Posições de convergência encontradas pelo StratVision - Pretas - Experimento 2



Figura 5.9: Cenários similares encontrados pelo StratVision - Experimento 2

8.830.762 posições. Em virtude, que um HLP deve ser analisado em posições do grafo-OU da árvore de busca, apenas as posições onde o lado que busca o padrão tem a vez do lance são analisadas. A busca foi realizada em duas etapas, primeiro como brancas e em seguida como pretas.

A análise de posições de partidas reais garante que são atingíveis através de movimentos válidos, e provém de caminhos distintos entre todas as possibilidades que a

árvore de busca do jogo proporciona. Isto equivale a uma simulação de busca durante uma partida real usando um algoritmo como MCTS ou MINIMAX.

Os resultados resumidos de cada lote estão na tabela 5.3. A tabela contém o total de partidas, número de posições processadas, número de ocorrências do cenário, número de ocorrências positivas do Philidor. O percentual do cenário foi calculado em função do total de posições avaliadas. O percentual do padrão foi calculado em função do número de posições onde o cenário foi encontrado.

Tabela 5.3: Resultados - Busca na base chess-db

Lote	Partidas	Posições	Cenário	%	Padrão	%
p1	14726	1190571	3161	0,266	3	0,095
p2	14733	1160902	3066	0,264	0	0,000
p3	14741	1088737	2718	0,248	1	0,037
p4	14737	1064540	2640	0,257	1	0,038
p5	14742	1109027	2871	0,259	4	0,139
p6	14744	1100581	3081	0,280	4	0,130
p7	14742	1072809	2841	0,265	5	0,176
p8	14745	1043595	2669	0,256	0	0,000
Total	117910	8830762	23047	0,261	18	0,078

Os resultados destes experimentos encontram-se divididos em casos de verdadeiros positivos (seção 5.1.1), falsos positivos (seção 5.1.1) e falsos negativos (seção 5.1.1) retornados pelo StratVision.

Verdadeiros positivos A tabela 5.4 contém as posições em formato FEN encontradas neste experimento e a avaliação estática calculada pela máquina Stockfish.

Tabela 5.4: Posições positivas - experimento 3 - Padrão Philidor

Lote	FEN	Stockfish
p1	Fig. 5.10 P01	5.03*
p1	Fig. 5.10 P02	-2.04
p3	Fig. 5.10 B01	1.25
p5	Fig. 5.10 B02	-0.08*
p5	Fig. 5.10 B03	-2.88*
p6	Fig. 5.10 B04	0.83
p7	Fig. 5.10 B05	1.52
p7	Fig. 5.10 B06	-2.62*
p7	Fig. 5.10 B07	1.52
p7	Fig. 5.10 P03	-2.26

Falsos positivos As posições da tabela 5.5 são falsos positivos retornados pela representação adotada. O problema das posições p4 e p7 recai sobre a casa S2 defendida pela Dama. Para resolver esta situação foi adicionada a seguinte cláusula condicional nas pré-condições:



Figura 5.10: Verdadeiros positivos - Experimento 3

"PIECESDEF(S2,q) = 0", ou seja, o número de Damas do oponente que defendem "S2" deve ser zero. Nos demais casos, exceto p6a e p6c, o problema acontece na casa "S1" que está protegida pela Torre, e a casa "S2" é protegida tanto pela Torre quanto pelo Cavalo, assim foi alterada a sentença "(DIRECTDEFENSES(S2,OPPONENT) = 2)" para "(DIRECTDEFENSES(S2,OPPONENT) = 2 AND (DIRECTDEFENSES(S1,OPPONENT) = 0 OR PIECESDEF(S1,N) = 2) AND PIECESDEF(S2,q) = 0)". Esta alteração garante também que a posição do jogo entre NN x Terry McCracken (tabela 5.1) não seja dada como falso negativo. A listagem 5.3 mostra como ficaram as pré-condições após o incremento condicional.

O caso em destaque da tabela 5.5 não foi resolvido com as alterações. Com estas

Tabela 5.5: Falsos positivos - experimento 3 - Padrão Philidor

Lote	FEN	Stockfish
p1	Fig. 5.11 B01	2.17
p4	Fig. 5.11 B02	0.42
p5	Fig. 5.11 B03	2.09
p5	Fig. 5.11 P01	1.17
p6a	Fig. 5.11 B04	3.22
p6b	Fig. 5.11 P02	2.04
p6c	Fig. 5.11 B05	0.75
p7	Fig. 5.11 P03	-0.55

modificações os falsos positivos foram reduzidos em 87,5%. A posição B04 apesar de ter um xeque-mate, não se trata do padrão Philidor, onde a Dama deveria ser sacrificada.



Figura 5.11: Falsos positivos - Experimento 3

```

1 PRECONDITION
2 (DIRECTDEFENSES(S2,OPPONENT) = 1 OR
3 (DIRECTDEFENSES(S2,OPPONENT) = 2 AND
4 (DIRECTDEFENSES(S1,OPPONENT) = 0 OR
5 PIECESDEF(S1,N) = 2) AND
6 PIECESDEF(S2,q) = 0)) AND
7 (DIRECTDEFENSES(S1,OPPONENT)= 0 OR
8 RELDEFENSESVALUE(S1,OPPONENT)=RELATIVEVALUEOF(f1)) AND
9 MOVESOF(k1) = 0 AND
10 (TYPEOF(F1,Q) OR
11 TYPEOF(F1,R)) AND
12 (TYPEOF(f1,R) OR
13 TYPEOF(f1,N))

```

Listagem 5.3: Pré-condição para evitar os falsos positivos encontrados

Falsos negativos A verificação de falsos negativos foi realizada por amostragem estatística usando o método das proporções para calcular o tamanho da amostra com 95% de confiança e erro amostral máximo de 5%, com uma estimativa de 0,5 para a variância, o que aumenta o tamanho da amostragem. Após definido o tamanho da amostra, os indivíduos foram selecionados aleatoriamente em todos os 8 lotes e inspecionados visualmente um a um, com apoio de enxadristas. Não foram encontrados casos de falsos negativos.

Experimento 4

Com as correções propostas, surgiu uma pergunta, seria possível detectar todos os casos do experimento 5.1.1 sem incluir as cláusulas TACTICALS e POSTCONDITION? Para responder esta questão, as cláusulas de táticas e pós-condições foram excluídas e o experimento foi executado novamente. O padrão foi encontrado corretamente em todas as posições. Também foram verificadas as posições verdadeiras do experimento 3 na ferramenta de autoria. O padrão foi encontrado corretamente em todas elas. A vantagem da remoção é o ganho de desempenho na busca. Importante dizer que para este padrão a remoção das cláusulas foi bem sucedida. Para outros padrões será necessário refazer este processo a fim de avaliar se não ocorrerão falsos negativos.

Resumo dos experimentos - Philidor

O padrão Philidor é um caso onde, uma vez que ele surge, existe uma sequência forçada de lances que terminam com um xeque-mate, sem que o oponente consiga reagir. Compreender seu funcionamento e todas as condições que devem ser verificadas somente é possível através de um processo incremental. Várias versões deste padrão foram implementadas. Três delas encontram-se descritas neste texto.

Os experimentos desta seção tiveram por objetivo testar a capacidade de representação do modelo adotado, medindo sua eficácia. Os experimentos foram divididos em 4 seções:

1. Seção 5.1.1: realizado de forma preliminar foram usadas as posições dadas por [40]. Em ambas o StratVision detectou o padrão.
2. Seção 5.1.1: O segundo experimento foi executado sobre uma relação de partidas conhecidas, onde 27 partidas chegaram à posição de convergência, e em 13 o lado perdedor desistiu antes de chegar na posição de convergência. O StratVision identificou corretamente o padrão nas 27 posições.
3. Seção 5.1.1: uma base de jogos com 117.910 partidas e mais de 8 milhões de posições foi usada. Ocorreram 8 casos de falsos positivos. Alterações foram feitas

no código PATLAN, reduzindo esse número para 1. Novos testes foram realizados em todas as posições dos demais experimentos sem alterações nos resultados.

4. Seção 5.1.1: por último foi executado um teste em todas as posições do Experimento 2 e do Experimento 3 para verificar a hipótese de que apenas as cláusulas SCENARIO e PRECONDITION são suficientes para representar o conhecimento do Philidor para realização de buscas. O experimento comprovou esta hipótese.

5.1.2 Sequencia não forçada - Xeque-mate Anastasia

O padrão Anastasia, bem conhecido pelos enxadristas experientes, foi representado usando a linguagem PATLAN. A ideia deste mate é combinar um ataque com Torre e Cavalo, tendo o rei adversário em uma das bordas com seus movimentos limitados. Existe uma peculiaridade neste padrão que é a possibilidade de alguma peça adversária defendida e até mesmo amiga se interpor no caminho entre a Torre e o Rei e frustrar o ataque. A figura 5.12 é um exemplo de quando uma peça amiga provoca interferência no padrão.

Embora o padrão exista, o fato do Bispo Preto estar presente na casa por onde a Torre preta deveria atacar o rei, faz com que o padrão se perca, pois o Rei pode se mover para h3 frustrando o ataque, ou o Peão se mover de g2 para g3, possibilitando um escape para o Rei. Para impedir que este caso seja dado como positivo foi colocada uma condição de que a casa para onde a Torre deve se mover tem que estar vazia. Se o objetivo de quem estiver representando é detectar apenas a similaridade com o padrão de Anastasia esta condição pode ser removida.

A figura 5.13 é um exemplo de quando uma interposição de uma peça adversária frustra a execução do padrão. A Dama branca pode se interpor entre a Torre preta e o Rei, frustrando a sequencia. Assim é necessário checar um conceito chamado nesta tese de interposição adversária com a função INTRUSOSENTRE (*WEDGESBETWEEN*) caso se queira impedir tais cenários. A não utilização desta função também permite a detecção de cenários similares ao Anastasia, porém sem garantias de execução do mesmo.

O que foi exposto até aqui justifica o fato deste padrão ser classificado como sendo de uma sequencia não forçada de lances.

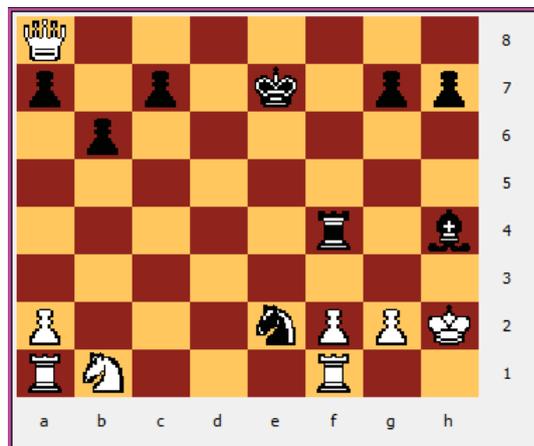


Figura 5.12: Cenário Anastasia - falso positivo amigo

Existem dois cenários onde o padrão Anastasia pode ocorrer: (1) através de um ataque possível ao Rei e (2) através de um ataque indireto ao Rei.

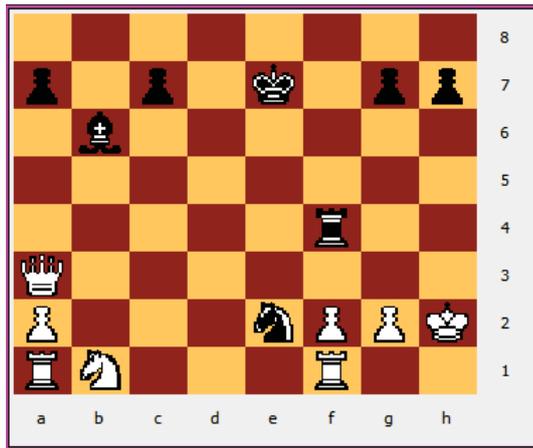


Figura 5.13: Cenário Anastasia - falso positivo interposto

O segundo cenário é quando existe uma interposição de uma peça **não defendida**, atrasando a derrota em um lance. Assim, duas representações foram feitas, sendo uma para cada cenário, como descritas nas listagens 5.4 e 5.5.

```

1 PATTERN "ANASTASIA_SCENARIO_01"
2 AUTHOR "Generic Anastasia"
3 DESCRIPTION "Anastasia through possible attack"
4 WEIGHT 1.0
5 SCENARIO
6   N1 DEFENDS S1
7   k1 DEFENDS S1
8   R1 POSSIBLEATTACK(>s2<,k1)
9   R1 POSSIBLEDEFENSE(s2,S3)
10  k1 DEFENDS S3
11  k1 DEFENDS p1
12 PRECONDITION
13   SQUAREOWNER(s2,NONE) AND
14   (RANKOF(N1) = RANKOF(p1) OR
15    FILEOF(N1) = FILEOF(p1)) AND
16   MAXDISTANCE(N1,p1) = 2 AND
17   WEDGESBETWEEN(s2,k1) = 0 AND
18   ONBORDER(k1) AND
19   ONBORDER(p1) = FALSE AND
20   MOVESOF(k1) < 3 AND
21   (RANKOF(p1) = RANKOF(k1) OR
22    FILEOF(p1) = FILEOF(k1))
23 TACTICALS
24   R1,s2
25 POSTCONDITION
26   MOVESOF(k1) = 0 AND
27   DIRECTDEFENSES(s2,OPPONENT) = 0
28 ;

```

Listagem 5.4: Código PATLAN - Mate Anastasia via ataque possível

```

1 PATTERN "ANASTASIA_SCENARIO_02"
2 AUTHOR "Anastasia scenario 02"
3 DESCRIPTION "Anastasia through indirect attack"
4 WEIGHT 1.0
5 SCENARIO
6   N1 DEFENDS S1
7   k1 DEFENDS S1
8   R1 INDIRECTATTACK(>f2<,k1)
9   R1 INDIRECTDEFENSE(f2,S3)
10  k1 DEFENDS S3
11  k1 DEFENDS p1
12 PRECONDITION
13  (RANKOF(N1) = RANKOF(p1) OR
14   FILEOF(N1) = FILEOF(p1)) AND
15  MAXDISTANCE(N1,p1) = 2 AND
16  WEDGESBETWEEN(f2,k1) = 0 AND
17  ONBORDER(k1) AND
18  ONBORDER(p1) = FALSE AND
19  MOVESOF(k1) < 3 AND
20  (RANKOF(p1) = RANKOF(k1) OR
21   FILEOF(p1) = FILEOF(k1))
22 TACTICALS
23   R1,f2
24 POSTCONDITION
25  MOVESOF(k1) = 0 AND
26  DIRECTDEFENSES(f2,OPONENT) = 0
27 ;

```

Listagem 5.5: Código PATLAN - Mate Anastasia via ataque indireto

A tabela 5.6 apresenta uma explicação detalhada da listagem 5.4. A tabela 5.7 apresenta uma explicação detalhada da listagem 5.5.

Linhas	Descrição
1 a 4	Documentação
5 a 11	Descrição do cenário
6	Cavalo amigo defende uma casa S1
7	O rei oponente também defende a posição S1
8	Torre amiga pode atacar o Rei por uma casa s2 não defendida pelo oponente
9 e 10	Torre amiga defende uma casa S3 através da casa s2, que também é defendida pelo Rei oponente
11	Rei oponente está defendendo um peão
12 a 22	Conjunto de pré-condições
13	A casa s2 deve estar vazia para evitar a detecção de cenários com ataque possível da Torre através de uma peça amiga, como na posição exemplificada na figura 5.12, o que resultaria em um falso positivo, pelo atraso de um movimento, permitindo ao rei evadir-se da ameaça
14 e 15	A linha ou fila do Cavalo amigo deve ser a mesma do Peão oponente
16	Distância entre o Cavalo amigo e o Peão oponente deve ser igual a 2
17	Não deve haver possibilidade de interposição do oponente entre a casa s2 e a casa de seu Rei
18 e 19	O Rei oponente tem que estar na borda para o padrão acontecer, mas o Peão não pode
20	Movimentos do Rei oponente devem estar limitados em no máximo 2
21 e 22	Linha ou fila do Rei oponente deve ser a mesma do Peão oponente
23 e 24	Especificação da tática de execução do padrão
25 a 27	Especificação das pós condições pertinentes ao padrão. O Rei oponente deve estar sufocado e a casa s2 deve continuar sem defesas pelo oponente.

Tabela 5.6: Explicação detalhada da listagem 5.4

Linhas	Descrição
1 a 4	Documentação
5 a 11	Descrição do cenário
6	Cavalo amigo defende uma casa S1
7	O rei oponente também defende a posição S1
8	Torre amiga ataca indiretamente o Rei por uma peça f2 não defendida pelo oponente
9 e 10	Torre amiga defende uma casa S3 através da peça f2, que também é defendida pelo Rei oponente
11	Rei oponente está defendendo um peão
12 a 19	Conjunto de pré-condições
13 e 14	A linha ou fila do Cavalo amigo deve ser a mesma do Peão oponente
15	Distância entre o Cavalo amigo e o Peão oponente deve ser igual a 2
16	Não deve haver possibilidade de interposição do oponente entre a casa s2 e a casa de seu Rei
17 e 18	O Rei oponente tem que estar na borda para o padrão acontecer, mas o Peão não pode
19	Movimentos do Rei oponente devem estar limitados em no máximo 2
20 e 21	Linha ou fila do Rei oponente deve ser a mesma do Peão oponente
22 e 23	Especificação da tática de execução do padrão
24 a 26	Especificação das pós condições para que o padrão. O Rei oponente deve estar sufocado e a casa s2 deve continuar sem defesas pelo oponente.

Tabela 5.7: Explicação detalhada da listagem 5.5

Experimento 1

Este experimento foi executado usando as partidas classificadas pelo sítio <http://www.chessgames.com/perl/chesscollection?cid=1036807>, acesso em 06/11/2018. A tabela 5.8 apresenta os resultados. Em 5 das 12 partidas o padrão foi encontrado corretamente e em todas a posição obteve avaliação estática de derrota pela máquina *Stockfish* para o lado vencedor apesar do xeque-mate estar apenas 1 *ply* à frente. A posição P03 da figura 5.14 resultou em um falso negativo, em virtude da condição WEDGESBETWEEN estar definida na cláusula PRECONDITION. A posição foi rejeitada pois existe uma Dama que pode se interpor. Uma solução seria aceitar estes casos removendo esta condição, mas no momento não tem como garantir se o padrão se efetivará. Em 5 partidas, um dos jogadores desistiu antes do padrão se formar. Em 1 partida o formato do xeque-mate não combina com a descrição clássica do padrão Anastasia.

Tabela 5.8: Resultados da busca - Anastasia (Vencedores em negrito)

Jogo	Posição de convergência	StockFish
aronin x tolush 1950	Não confere com o padrão explícito de Anastasia	-
baron x marcelin 2013	Branças desistem antes do padrão surgir	-
bayer x falkbeer 1852	Fig. 5.14 P01	10,04
berryman x straat 1919	Fig. 5.14 B01	-8,88
dai x motycakova 2012	Fig. 5.14 P02	5,5
golmayo x loyd 1867	Fig. 5.14 P03	14,22
hammer x carlsen 2003	Branças desistem antes do padrão surgir	-
leonhardt x englund 1908	Branças desistem antes do padrão surgir	-
roth x rajna 1975	Fig. 5.14 B02	-3,48
vidmar x euwe 1929	Pretas desistem antes do padrão surgir	-
viner x carey 1961	Fig. 5.14 B03	-9,28
voronova x vo hong phuong 2000	Pretas desistem antes do padrão surgir	-

Experimento 2

Este experimento foi executado na mesma máquina e base de partidas do experimento 3 (seção 5.1.1) do xeque-mate Philidor. A tabela 5.9 apresenta os resultados da busca para o cenário da listagem 5.4. A tabela 5.10 mostra os resultados para o cenário da listagem 5.5, de forma similar à tabela do Philidor.

A busca não retornou dados para uma análise sobre falsos positivos. O procedimento adotado para verificação de falsos negativos é o mesmo que foi usado para o padrão Philidor. Também não foram encontrados casos de falsos negativos. Porém, como se trata de uma estimativa não é possível afirmar com 100% de confiança que não houveram casos de falsos negativos, tanto para o Anastasia como para o Philidor.

5.2 Representação de HLPs passivos

HLPs passivos são aqueles que não necessitam da formação de um cenário específico no qual as ações táticas envolvem elementos do cenário, mas apenas exigem prova de conceitos e nos quais as ações táticas estão relacionadas a estes conceitos. O exemplo



Figura 5.14: Posições encontradas - Experimento 3

Tabela 5.9: Resultados da busca - Anastasia - Cenário 01

Lote	Partidas	Posições	Cenário	%	Padrão	%
p1	14726	1190571	2819	0,237	0	0,000
p2	14733	1160902	2825	0,243	0	0,000
p3	14741	1088737	2494	0,229	0	0,000
p4	14737	1064540	2398	0,225	0	0,000
p5	14742	1109027	2622	0,236	0	0,000
p6	14744	1100581	2770	0,252	0	0,000
p7	14742	1072809	2726	0,254	0	0,000
p8	14745	1043595	2297	0,220	0	0,000
Total	117910	8830762	20951	0,237	0	0,000

dados no capítulo 1 na figura 1.2 (página 21) é um caso típico. O cenário em si não define algum padrão característico, talvez, com exceção da estrutura de peões das peças pretas que parecem formar uma barreira. A visão estratégica do vencedor lhe permitiu realizar a "Caminhada do Rei", isto é, pela percepção de que havia um caminho seguro do Rei até o Peão oponente próximo ao rei adversário.

5.2.1 HLP - Rei pressionado por peão passado

Do trabalho de Linhares [40] as duas posições da figura 5.15 encontram-se nesta categoria de HLPs. A listagem 5.6 apresenta a codificação em PATLAN para uma possível identificação deste HLP em uma posição. A ideia destas duas posições é coroar o Peão passado com o apoio do Rei.

Apesar de HLPs passivos não dependerem de um cenário específico, a linguagem PATLAN exige que pelo menos uma ação esteja definida no cenário. Por esta razão foi definida a ação "P1 DEFENDS P2". A linha 5 deixa explícito que além dos Peões, ambos

Tabela 5.10: Resultados da busca - Anastasia - Cenário 02

Lote	Partidas	Posições	Cenário	%	Padrão	%
p1	14726	1190571	18	0,002	0	0,000
p2	14733	1160902	22	0,002	0	0,000
p3	14741	1088737	20	0,002	0	0,000
p4	14737	1064540	24	0,001	0	0,000
p5	14742	1109027	21	0,002	0	0,000
p6	14744	1100581	13	0,001	0	0,000
p7	14742	1072809	22	0,002	0	0,000
p8	14745	1043595	24	0,002	0	0,000
Total	117910	8830762	155	0,002	0	0,000

os Reis amigo e oponente devem estar presentes. A linha 6 define em qual contexto este padrão deve ser buscado, ou seja, em finais de Rei e Peão.

A função "EXISTEPADRAO" ("*EXISPATTERN*") permite verificar se na mesma posição existe algum outro padrão. No caso da listagem é verificado se não existe um padrão de nome "PASSEDPAWN" ou "PEAOPASSADO" no lado do oponente. Um Peão é considerado passado quando ele tem caminho livre para promoção, sem Peões oponentes atuando como sentinelas ou como bloqueadores.

A cláusula PRECONDITION define as seguintes condições:

- A linha 10 e 11 indicam que P2 não deve ter Peões oponentes atuando como sentinelas e nem bloqueadores que impeçam seu avanço para a promoção.
- A linha 12 indica utiliza um conceito enxadrístico conhecido como "Regra do Quadrado", neste caso o Rei oponente deve estar no mesmo quadrado do Peão passado, assim ele fica ocupado tentando impedir a promoção enquanto o Rei amigo poderá executar sua caminhada para dar suporte ao coroamento do Peão.
- A linha 13 indica que deve haver um caminho seguro entre o Rei amigo e o Peão passado para que ele possa realizar a caminhada.
- A linha 14 introduz o conceito denominado de oposição de Reis (KINGOPPOSITION). A oposição ocorre quando o Rei de um jogador impede o avanço do outro Rei. Para este HLP não deve existir uma oposição de Reis.

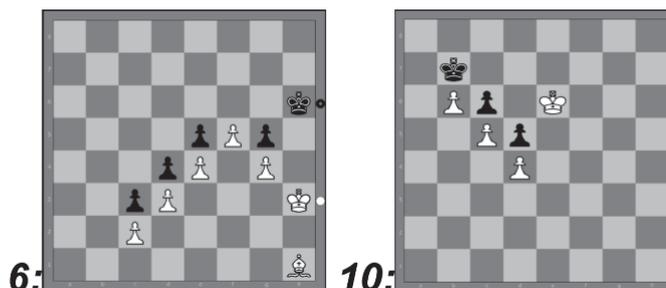


Figura 5.15: Exemplo de HLP passivo - Fonte:[40]

```

1 PATTERN "LINHARES_06_10"
2 AUTHOR "Luis Bueno"
3 DESCRIPTION "Passed pawn pushing opponent king"
4 WEIGHT 1.0
5 ACTORS K1,k1
6 SETOFPIECES KPkP
7 SCENARIO
8   P1 DEFENDS P2
9 PRECONDITION
10  SENTRY(P2) = 0 AND
11  RAM(P2) = 0 AND
12  INTHE SQUARE(k1,P2) AND
13  KINGPATHTO(K1,P2) > 0 AND
14  KINGOPPOSITION(K1,k1) = FALSE AND
15  EXISTPATTERN("PASSEDPAWN",OPPONENT) = FALSE;

```

Listagem 5.6: Código PATLAN - HLP Passivo - Linhares 06 e 10

Experimento 1

Os experimentos realizados nas posições da figura 5.15 foram positivos. Para a posição 06, entretanto, é necessário acrescentar o Bispo branco no conjunto de peças do padrão. O StratVision reconheceu que os conceitos representados se aplicam a ambos os casos.

Experimento 2

O experimento 2 foi realizado usando os mesmos 8 lotes da base de partidas do Philidor e Anastasia. Todas as posições de cada partida foram geradas e testadas para verificar a existência do HLP passivo em questão.

É importante lembrar que antes do StratVision fazer o casamento do cenário e a prova das condições, é feita a análise do conjunto mínimo de material exigido e/ou a exigência de um conjunto exclusivo de peças. Caso uma posição não tenha o conjunto mínimo e/ou exclusivo, o StratVision avança para a próxima posição da partida em análise. Para fins de agilizar a busca o cenário foi restrito apenas ao conjunto exclusivo de Reis e Peões.

A tabela 5.11 apresenta o resumo da busca nos 08 lotes do HLP da listagem 5.6.

Tabela 5.11: Resultados da busca - Anastasia - Cenário 02

Lote	Partidas	Posições	Cenário	%	Padrão	%
p1	14726	1190571	2207	0,185	33	1,495
p2	14733	1160902	1651	0,142	30	1,817
p3	14741	1088737	1555	0,143	21	1,350
p4	14737	1064540	1529	0,144	28	1,831
p5	14742	1109027	1538	0,139	32	2,081
p6	14744	1100581	1800	0,164	20	1,111
p7	14742	1072809	1596	0,149	37	2,318
p8	14745	1043595	1558	0,149	14	0,899
Total	117910	8830762	13434	0,152	215	1,600

Por razões de espaço, foram transcritas algumas das posições encontradas no lote "p1". A tabela 5.12 contém os casos de verdadeiros positivos. A primeira parte refere-se às posições onde as brancas tem chances de vitória e a segunda onde as pretas tem chances de vitória. A tabela 5.13 apresenta os falsos positivos encontrados no lote

Tabela 5.12: Verdadeiros positivos - experimento 2

Pos	Verdadeiros positivos	Stockfish
1	Fig. 5.16 B01	3.38
2	Fig. 5.16 B02	4.41
3	Fig. 5.16 B03	1.58
4	Fig. 5.16 B04	2.02
5	Fig. 5.16 B05	1.95
6	Fig. 5.16 B06	1.25
7	Fig. 5.16 B07	0.99
8	Fig. 5.16 B08	1.27
9	Fig. 5.17 P01	-1.16
10	Fig. 5.17 P02	-1.44
11	Fig. 5.17 P03	-1.6

Tabela 5.13: Falsos positivos - experimento 2

Pos	Falso positivos	Stockfish
1	Fig. 5.18	-2.01

"p1", coincidentemente, as partidas onde foram encontradas terminaram empatadas. Pelos cálculos de avaliação estática da máquina Stockfish, poucas posições apresentam vantagem decisiva (acima de 2 peões) para o lado que joga (1, 2, 6, 18 e 31).

Outro aspecto a ser observado pela listagem 5.6 é que neste padrão não existem táticas definidas pois podem exigir muitos passos táticos para atingir a vantagem estratégica percebida.

Um comportamento observado é que um HLP passivo uma vez encontrado, pode se perpetuar por vários movimentos durante a partida e o StratVision manteve a percepção correta. Um exemplo disto pode ser visto nas posições 9 a 11.

As demais posições do trabalho de Linhares [40] foram representadas (vide o apêndice D) e testadas com as posições propostas em seu trabalho, com eficácia de 100%. Entretanto, os principais esforços foram realizados nos experimentos relatados neste capítulo por serem mais significativos e complexos para representar.

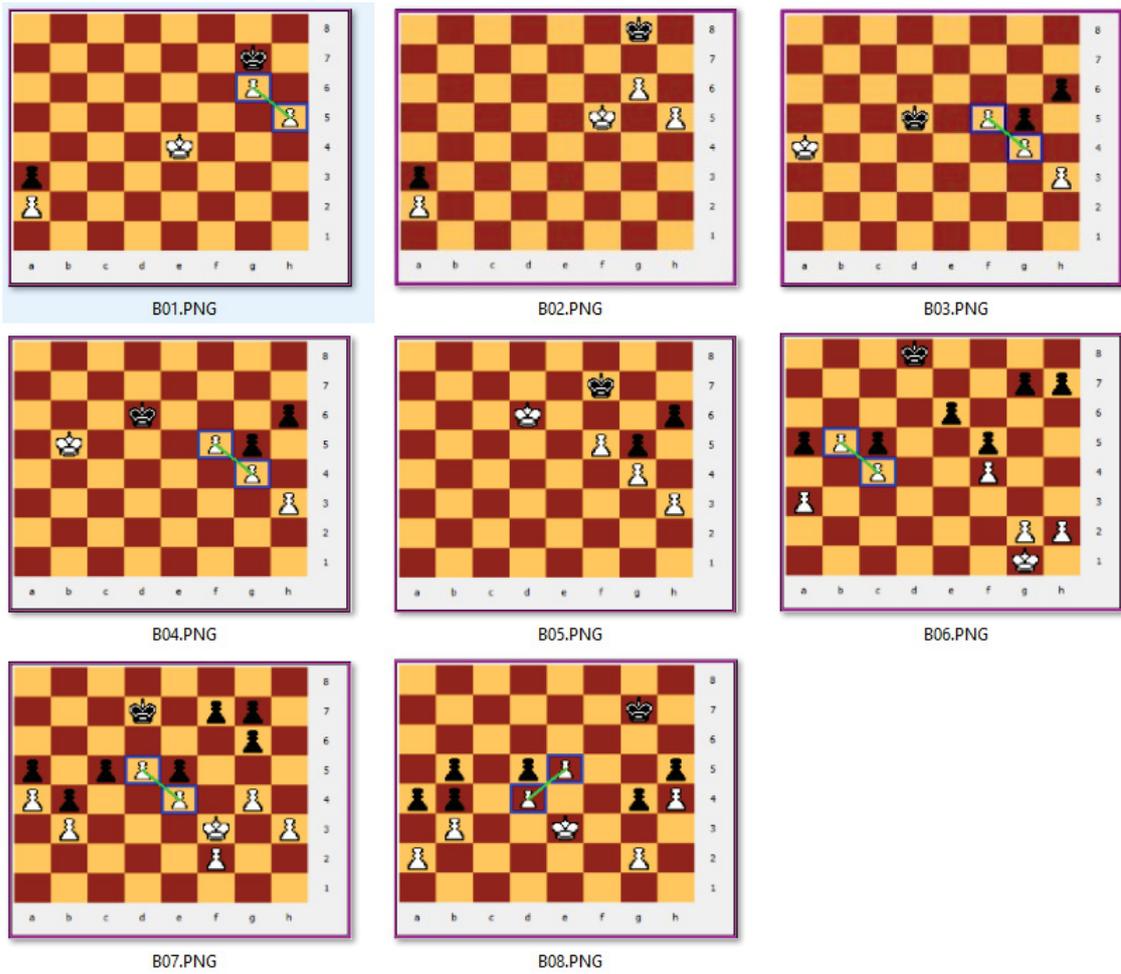


Figura 5.16: Exemplo de posições positivas encontradas - HLPs passivos - Brancas - Experimento 2

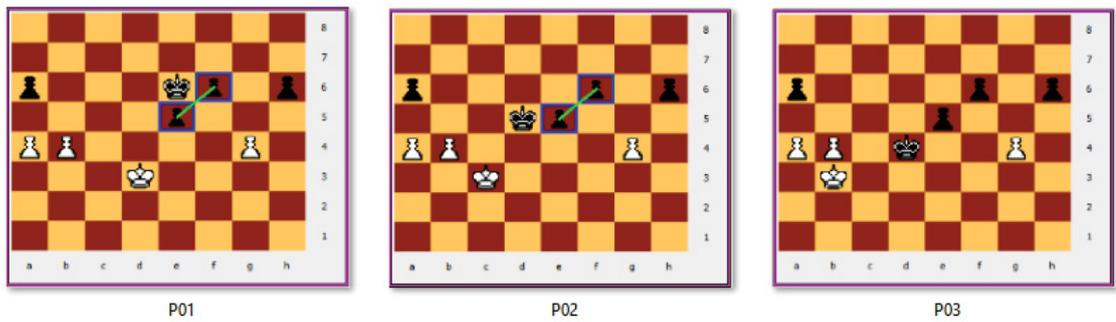


Figura 5.17: Exemplo de posições positivas encontradas - HLPs passivos - Pretas - Experimento 2

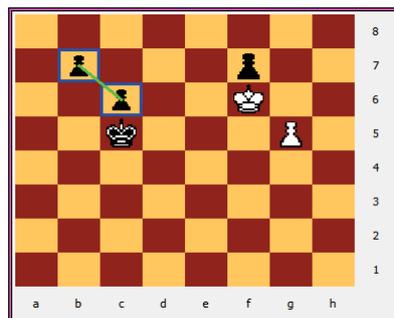


Figura 5.18: Falso positivo encontrado - HLPs passivos - Experimento 2

6 Considerações Finais

Este capítulo apresenta uma discussão sobre os experimentos e resultados obtidos, considerações finais sobre o projeto de pesquisa, limitações e trabalhos futuros.

6.1 Discussão

Duas hipóteses foram definidas nesta tese (seção 1.5): (1) é possível representar o conhecimento de padrões complexos no nível de visão estratégica e (2) é possível fazer a máquina perceber quando duas posições diferentes contém o mesmo padrão. Com base nos resultados dos experimentos, conclui-se que as hipóteses são verdadeiras.

A primeira hipótese é verdadeira porque é possível representar com sucesso tanto padrões ativos quanto passivos através do modelo ontológico proposto e da linguagem PATLAN. O padrão Philidor, por exemplo, precisa de 3 *plies* para chegar a posição de xeque-mate. Com a representação adotada, olhando apenas as consequências dos movimentos um nível abaixo na árvore de busca, é possível detectar se o padrão existe, conforme resultados do experimento 5.1.1. Isto gera uma redução significativa no custo da busca da ordem de 2 *plies*.

Um dos problemas conhecidos do algoritmo MINIMAX com avaliação estática é chamado de "Efeito do Horizonte". Berliner [6] considera-o como o nome dado a uma classe de aberrações que ocorre em algoritmos com busca em profundidade e largura. Em jogos como o xadrez, a expansão desta árvore de todos os estados é altamente custosa, por isso adota-se uma profundidade de corte quando não se encontra algum estado terminal, produzindo então estados terminais artificiais. Nestes estados aplica-se a FAE, qualquer coisa que não seja detectável no momento da avaliação não existe para o algoritmo, mesmo que seja um xeque-mate Philidor a 3 *plies* de distância. Por esta razão certos caminhos da busca podem ser ignorados por apresentarem valores controversos, ou seja, desvantagem para o lado que tem a posse do padrão.

A presença deste padrão pode ocorrer em profundidades relativamente baixas, isto é possível observar em algumas partidas usadas no experimento 5.1.1. Por exemplo, em uma das partidas entre NN x Grecco o padrão surgiu no lance 12 e no caso da partida entre Rousseau x Hume o padrão surgiu após o lance 13. Se uma máquina usar conhecimento de padrões para avaliar, é possível afirmar que: se além das habituais avaliações táticas, houver uma busca por padrões, os efeitos do problema do horizonte podem ser minimizados.

A segunda hipótese também é verdadeira, conforme comprovado pelos experimentos realizados. Todas as posições do trabalho de Linhares [40], obtiveram sucesso de reconhecimento pelos algoritmos do StratVision com as representações adotadas, ou seja, o algoritmo consegue identificar a similaridade no nível da visão estratégica através das instâncias positivas dos padrões.

O StratVision se mostrou eficaz nos casos citados no parágrafo anterior e também nos experimentos de busca em partidas onde já se sabia que os padrões buscados estavam presentes (Philidor e Anastasia). Porém ao realizar a busca relatada no experimento 5.1.1 ocorreram falsos positivos. Foram registrados, inicialmente, 8 casos de falsos positivos. Em termos do número total de cenários avaliados como similares às ações do padrão (23.047) os falsos positivos ocorreram em 0,034% dos casos. Com as mudanças efetuadas no código este valor foi reduzido para 0,004%.

Na tentativa de identificar falsos negativos no experimento 5.1.1 e considerando o alto número de cenários positivos, tornando-se inviável uma análise um a um, foi realizada uma amostragem estatística aleatória com tamanho da amostra igual a 380, com nível de confiança de 95% e margem de erro de 5%. Cada posição da amostra foi avaliada e não foram encontrados casos de falsos negativos.

Considerando o fato de existirem falsos positivos encontrados tanto no HLP de Philidor quanto no Anastasia, o StratVision pode ser usado como uma espécie de consultor estratégico em máquinas táticas, como StockFish, para sugerir a presença de padrões importantes estrategicamente, na tentativa de induzir o jogo para estas posições na árvore de busca. Uma vez alcançada a posição, uma máquina tática poderá concluir a tarefa avaliando se o padrão terá sucesso levando à vitória ou não.

Não foi realizado nesta pesquisa um estudo aprofundado sobre desempenho dos algoritmos do Stratvision, pelo fato de ter sido criado em linguagem Java que é interpretada, o que já apresenta desempenho inferior a código de máquina nativo. Entretanto alguns dados preliminares foram levantados. O que foi observado é que HLPs ativos, devido à complexidade de avaliação do cenário pela máquina TuProlog, apresentam desempenho bem inferior aos HLPs passivos. Além disto, algumas providências foram tomadas para melhorar o desempenho, tais como: redução do escopo da busca pela avaliação do material necessário e conjuntos exclusivos de peças.

O padrão Philidor ao ser executado apresentou um desempenho médio bruto de 295 posições/segundo. Em contrapartida, o desempenho do HLP passivo estudado chegou a 2725 posições/segundo de desempenho médio. Nestes cálculos estão inclusos: o tempo para ler cada partida em formato PGN, executar cada movimento no *bitboard*, validar o material e o conjunto exclusivo de peças quando for o caso, gerar a base de fatos, efetuar a busca do cenário na posição, estruturar as instâncias retornadas, avaliar as pré-condições, executar as táticas para gerar o estado final do HLP, e avaliar as pós-condições.

6.2 Impactos esperados

Espera-se com a presente tese que novas perspectivas venham a surgir, tanto para o campo da IA, quanto para outras áreas da ciência humana. Uma delas, em especial, seria a Medicina, na modelagem de conhecimento sobre doenças e diagnósticos baseado em padrões.

O problema dos falso positivos e falso negativos, ou seja, erros médicos no diagnóstico das doenças, foi estudado por Sonderegger-Iseli *et. al.* [71], que define o processo de diagnóstico como uma interação complexa de habilidades cognitivas e procedimentos técnicos em condições de incerteza. Os resultados do estudo que cobriu três décadas mostraram que houve uma queda pela metade nos erros, entre 1972 e 1992, principalmente nos casos de doenças cardiovasculares. Apesar disto, o artigo conclui, que estratégias para melhorar o desempenho do diagnóstico médico devem ser continuamente desenvolvidas.

Seria de grande valia realizar um estudo de viabilidade técnica para adaptar a

presente abordagem a modelos de diagnósticos médicos com base na visão estratégica. Uma ideia inicial de como isto poderia ser feito seria adaptar a ontologia e a linguagem PATLAN da seguinte forma:

- A cláusula CENARIO seria mapeada para o quadro clínico atual de um paciente.
- Os atores e ações seriam mapeados para os sintomas, exemplo: um ator poderia ser um órgão específico do corpo humano e uma ação seria o sintoma (dor, aperto, tosse, febre, sangramento etc.)
- A cláusula PRECONDIÇÃO poderia ser definida como os exames e resultados necessários para avaliar aquele estado clínico. Se estiverem de acordo com o esperado pelo padrão passa-se para a próxima fase, a profilaxia.
- A cláusula TACTICALS pode ser mapeada para a profilaxia indicada. Assim como no xadrez, as táticas se referem ao elemento temporal de um HLP, o tratamento terapêutico é o elemento temporal de um diagnóstico. Ao término de um intervalo específico de tempo, passa-se para o próximo passo.
- A cláusula POSTCONDITION pode ser mapeada como sendo os exames necessários e os valores esperados após o tratamento no caso do padrão da doença estar correto.

A abordagem desta tese poderia também ser adaptada para o modelo de conteúdo em STIs para treinamento médico no diagnóstico de enfermidades. Além da medicina, outras áreas também podem se beneficiar, como administração de empresas, de forma análoga a medicina, voltado por exemplo para jogos de empresas na avaliação de cenários estratégicos. Outras aplicações ainda poderiam ser:

- Geração de uma base de conhecimento sobre HLPs, como complemento aos livros de abertura e as bases de finais de jogo;
- Multi-escalamento de HLPs passivos, onde um HLP passivo seria um estado futuro a ser atingido a partir de um outro HLP. Como resultado se teria uma rede de conexões entre padrões. Estas conexões poderiam, por exemplo, expressar causalidade e efeito, ou seja, padrões que são consequentes de outros padrões.
- Desenvolvimento de STIs de jogos adversaristas baseados em padrões para aquisição de habilidades no nível de visão estratégica com potencial de amplificação do impacto que os computadores tem sobre a comunidade enxadrística na formação de competidores humanos (jogadores em geral) que aprenderam desde cedo a jogar com as máquinas.

Também espera-se que seja útil para a comunidade enxadrística no mundo, apoiando na análise de partidas, na busca por posições específicas e na própria autoria de padrões que ainda não estão documentados.

6.3 Conclusão

A presente tese foi fundamentada na hipótese central de que é possível formalizar o conhecimento no nível de visão estratégica, que permite a um humano identificar

similaridades em cenários aparentemente diferentes. Como domínio de testes foi adotado o jogo de Xadrez. Grandes mestres enxadristas possuem a capacidade de armazenar milhares de padrões em sua memória e estabelecem suas estratégias a partir de tal habilidade.

Para comprovar esta hipótese várias atividades foram executadas, dentre elas:

- Estudo de padrões enxadrísticos.
- Definição de uma ontologia sobre padrões, aqui denominados de HLPs.
- Criação da linguagem PATLAN para formalização de conhecimento de padrões.
- Criação do ambiente StratVision composto de: ferramenta de autoria, parser e interpretador da linguagem PATLAN, módulo de casamento de padrões, avaliação de instâncias e visualização da busca usando MREs.
- Representação de HLPs e experimentos de busca.

6.3.1 Limitações

Apesar dos bons resultados obtidos, existem limitações que ainda precisam ser resolvidas, entre elas destacam-se:

- Desempenho lento devido às tecnologias utilizadas;
- Impossibilidade de interpretar ações dinâmicas com mais de dois atores passivos;
- Falhas nas instanciações quando se usa disjunção de ações no cenário. No caso do xeque-mate Anastasia, foi necessário criar dois HLPs específicos devido à esta limitação. No momento disjunções de ações devem ser usadas com critério.
- Pela análise das partidas usadas nos experimentos, percebeu-se que humanos conseguem se antecipar e detectar que um HLP ativo inevitavelmente irá ocorrer vários lances antes da posição de convergência surgir. Nas partidas entre Alexander Alekhine x Menzel e Arthur William Dake x Jim Schmitt (disponíveis em <http://www.chessgames.com/perl/chesscollection?cid=1001808> - acesso em 18/11/2018), os perdedores desistiram 7 movimentos antes. O StratVision não é capaz disso.

6.3.2 Trabalhos futuros

A seguir, estão definidos alguns trabalhos para a continuidade desta pesquisa:

- Melhorias no interpretador PATLAN para tratar ações dinâmicas com mais de dois atores passivos que permitam, por exemplo, detectar raios-x de longo alcance.
- Expansão e aprimoramento dos conceitos pré-definidos da linguagem PATLAN. Um exemplo é a função WEDGESBETWEEN (INTRUSOSENTRE). Para evitar falsos negativos em HLPs semelhantes ao Anastasia será necessário uma função que retorne o número de intrusos que ao se moverem para a casa alvo, ficam defendidas por outra peça amiga.

- Criação de uma máquina que inclua o reconhecimento de HLPs durante a execução de uma partida, após a avaliação estática, como uma espécie de consultor estratégico, que altere o valor heurístico retornado, possibilitando provocar uma perturbação no cálculo tático, a fim de selecionar um movimento com viés mais estratégico.
- Criar heurísticas de longo prazo que permitam estimar a distância/custo em lances que uma posição qualquer encontra-se de um HLP. Uma possível heurística seria: considerando-se inicialmente o tabuleiro somente com as peças participantes da ação, calcular o somatório das distâncias euclidiana, ou Manhattan, entre a casa atual da peça e a casa onde ela deve estar para atingir o ponto de convergência do padrão.
- Incorporar o ambiente StratVision aos ambientes de competição alternada com colaboração já desenvolvidos pelo Departamento de Informática da UFPR.
- Na seção 2.6 foi revisto o trabalho AlphaGo Zero que usa o algoritmo MCTS ao invés do algoritmo MINIMAX, no campo do aprendizado de máquina. Fazer um estudo de viabilidade técnica para criar máquinas que se tornem especializadas em certos tipos de xeque-mate.
- Verificar o impacto desta abordagem no ensino de xadrez.
- Realizar um estudo de viabilidade para adaptar o ferramental aos domínios de medicina e administração como sugerido na seção 6.2.

Referências

- [1] Shaaron Ainsworth. The functions of multiple representations. *Computers & Education*, 33:131–152, 1999.
- [2] Shaaron AinsWorth. Deft: A conceptual framework for considering learning with multiple representations. *Learning and Instruction*, 16(3):183–198, 2006.
- [3] Shaaron E. Ainsworth, Peter A. Bibby, and David J. Wood. Information technology and multiple representations: new opportunities - new problems. *Information Technology for Teacher Education*, 6(1):93–104, 1997.
- [4] Maria Angeles Andreu-andre and Miguel Garcia-Casas. Perceptions of Gaming as Experiential Learning by Engineering Students. *International Journal of Engineering Education*, 27(4):795–804, 2011.
- [5] Eric D. Beinhocker. *The Origin of Wealth: Evolution, Complexity and the Radical Remaking of Economics*. Harvard Business School Press, Boston, MA, 2007.
- [6] Hans J Berliner. Some Necessary Conditions for a Master Chess Program. In *Proceedings of the Third International Joint Conference on Artificial Intelligence.*, Stanford, CA, USA., 1973. Stanford University.
- [7] Guinness Book. *Most moves in a chess game*, 2018 (acessado em 06/12/2018).
- [8] I Bratko, D Kopec, and D Michie. Pattern-Based Representation of Chess End-Game Knowledge. *Computer Journal*, 21(c):149–153, 1978.
- [9] C B Browne, E Powley, D Whitehouse, S M Lucas, P I Cowling, P Rohlfshagen, S Tavener, D Perez, S Samothrakis, and S Colton. A Survey of Monte Carlo Tree Search Methods. *Computational Intelligence and AI in Games, IEEE Transactions on*, 4(1):1–43, 2012.
- [10] Bernd Brugmann. Monte Carlo Go. Technical Report ii, Max-Planke - Institute of Physics, Universidade de Siracusa, 1993.
- [11] Luis; Bueno, Alexandre; Direne, André; Guedes, Fabiano; Silva, Luis; Bona, Laura García, Marcos; Castilho, and Marcos; Sunyé. Conceitos e ferramentas de apoio ao aperfeiçoamento do desempenho de heurísticas de jogos. In *Anais do XXVIII Congresso da Brasileira da Educação - SBC, Workshop sobre Informática na Escola*, 2008.
- [12] Jaime R Carbonell. AI in CAI: An Artificial-Intelligence Approach to Computer-Assisted Instruction. *Ieee Transactions on Man-Machine Systems*, 11(4):190 – 202, 1971.
- [13] William G. Chase and Herbert a. Simon. Perception in chess. *Cognitive Psychology*, 4(1):55–81, 1973.

- [14] Cristina Conati. Intelligent Tutoring Systems: New Challenges and Directions. *Science*, pages 2–7, 2002.
- [15] Wilson da; Silva. *Raciocínio lógico e o jogo de xadrez: em busca de relações*. PhD thesis, Universidade Estadual de Campinas, 2010.
- [16] O. David-Tabibi, H. Jaap van den Herik, M. Hoppel, and N. Netanyahu. Simulating human grandmasters: evolution and coevolution of evaluation functions. In *GECCO'09 Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*, pages 1483–1490, 2009.
- [17] Randall Davis, Howard Shrobe, and Peter Szolovits. What is a knowledge representation? *AI Magazine*, 14(1):17–33, 1993.
- [18] Adriaan De Groot. *Thought and Choice in Chess*. Mouton Publishers, New York, 2a. edition, 1978 (1946).
- [19] Edward Deci, Robert Vallerand, Luc Pelletier, and Richard Ryan. Motivation and Education: The Self-Determination Perspective. *Educational Psychologist*, 26(3):325–346, 1991.
- [20] Enrico Denti, Andrea Omicini, and Alessandro Ricci. tuProlog: A Light-weight Prolog for Internet Applications and Infrastructures. *Practical Aspects of Declarative Languages. 3rd International Symposium (PADL 2001), Las Vegas, Nevada, March 11–12, 2001 Proceedings*, 1990:184–198, 2001.
- [21] Alexandre Direne, Luis Bona Bona, FabFiano Silva Silva, Gabriel dos Santos, André Guedes, Marcos Castilho, Marcos Sunyé, Celso Hartmann, Pedro de Andrade Neto, Samuel Mello, Jaime Sunyé Neto, and Wilson da Silva. Conceitos e ferramentas de apoio ao ensino de xadrez nas escolas brasileiras. In SBC, editor, *Anais do XXIV Congresso da Sociedade Brasileira de Computação: WIE - Informática na Escola*, pages 816–825, Salvador - Brasil, 2004.
- [22] Alexandre; Feitosa, Alexandre; Direne, Fabiano; Silva, Luis; Bona, and Wilson da; Silva. Simulated Competitions to Aid Tactical Skill Acquisition. In *Winter Simulation Conference - WSC2013, 2013, Washington - EUA. Proceedings of the 2013 Winter Simulation Conference - WSC2013.*, pages 3588–3599. Catonsville: Simulation Society, v. 1, 2013.
- [23] Alexandre Rômolo Moreira Feitosa. *Definição Formal de Táticas de Xadrez por Meio da Autoria Incremental de Conceitos Heurísticos. Dissertação de Mestrado*. PhD thesis, Universidade Federal do Paraná- Brasil., 2006.
- [24] L Finkelstein and S Markovitch. Learning to play chess selectively by acquiring move patterns. *ICCA Journal*, 21:100–119, 1998.
- [25] Dinesh Gadwal, Jim E Greer, and Gordon I McCalla. Tutoring bishop-pawn endgames: An experiment in using knowledge-based chess as a domain for intelligent tutoring. *Applied Intelligence*, 3(3):207–224, 1993.
- [26] Fernand Gobet and Guillermo Campitelli. Educational benefits of chess instruction: A critical review. *Chess and Education: Selected essays from the Koltanowski Conference*, 44(115):230, 2006.

- [27] C Hartmann, A Direne, and L Bona. Linguagem E Ferramenta De Autoria Para Promover O Desenvolvimento De Perícias Em Xadrez. In *XVI Simpósio Brasileiro de Informática na Educação - SBIE*, pages 645–654, 2005.
- [28] Marcel Herbst. Chess Metaphors: Artificial Intelligence and the Human Mind. *The European Legacy*, 20(7):787–789, 2015.
- [29] Albert E Hickey and John M Newton. Computer-Assisted Instruction: A Survey of The Literature. Technical report, Office of Naval Research, Newburyport, Massachusetts, 1967.
- [30] Marko A. Hofmann and Bodo Junge. Dealing with structural uncertainty in tactical wargaming. In *Proceedings of the 2008 Summer Computer Simulation Conference, SCSC '08*, pages 10:1–10:13, Vista, CA, 2008. Society for Modeling & Simulation International.
- [31] Johan Huizinga. *Homo Ludens*. Editora Perspectiva, São Paulo, SP, Brasil, 2000.
- [32] H. Kaindl. Positional Long Range Planning Computer Chess. *Advances in Computer Chess*, 3:145–168, 1982.
- [33] Richard Korf. Planning as Search: A Quantitative Approach. *Artificial Intelligence*, 33:65–88, 1987.
- [34] Alexander Kotov. *Juague como un gran maestro*. Club de Ajedrez, 1989.
- [35] Eleandro Maschio Krynski and Alexandre Ibrahim Direne. Múltiplas representações externas para o ensino de programação de computadores. In *Anais do I Workshop de Ambientes de apoio à aprendizagem de algoritmos e programação do SBIE-2007*, volume 1, pages 1–8(CDROM), São Paulo, 2007.
- [36] N. Lassabe, S. Sanchez, H. Luga, and Y. Duthen. Genetically programmed strategies for chess endgame. In *GECCO'06 Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, pages 831–838, 2006.
- [37] Santos Gerardo Lazzeri and Rachelle Heller. Iconchess: An interactive consultant for chess middlegames. In *Proceedings of the 1996 International Conference on Learning Sciences, ICLS '96*, pages 456–461. International Society of the Learning Sciences, 1996.
- [38] Krittaya Leelawong and Gautam Biswas. Designing Learning by Teaching Agents: The Betty's Brain System. *International Journal of Artificial Intelligence in Education*, 18:181–208, 2008.
- [39] Alexandre Linhares. The emergence of choice: Decision-making and strategic thinking through analogies. *Information Sciences*, 259:36–56, 2014.
- [40] Alexandre Linhares and Paulo Brum. Understanding our understanding of strategic scenarios: what role do chunks play? *Cognitive science*, 31(6):989–1007, 2007.
- [41] Collin F Lynch, Kevin D Ashley, Vincent Aleven, and Niels Pinkwart. Defining “ Ill-Defined Domains ”; A literature survey . *Proceedings of the Workshop on Intelligent Tutoring Systems for IllDefined Domains at the 8th International Conference on Intelligent Tutoring Systems*, pages 1–10, 2006.

- [42] Thomas V Malone. What makes things fun to learn? heuristics for designing instructional computer games. *Proceedings of the 3rd ACM SIGSMALL symposium and the first SIGPC symposium on Small systems - SIGSMALL '80*, 162:162–169, 1980.
- [43] R. Marler. *101 Killer Chess Strategies*. CreateSpace Independent Publishing Platform, 2015.
- [44] T. Anthony Marsland and Yngvi Björnsson. *From MiniMax to Manhattan*. AAAI, 1997.
- [45] Daniel Martineschen, Alexandre I Direne, Luis Carlos De Bona, Fabiano Silva, and Marcos Castilho. Alternância Entre Competição E Colaboração Para Promover O Aprendizado Por Meio De Heurísticas De Jogos. In *WIE2006 - Workshop sobre Informática na Escola, 2006, Campo Grande. Anais do XXVI Congresso da Sociedade Brasileira de Computação. Porto Alegre.*, pages 50–58. Sociedade Brasileira de Computação., 2006.
- [46] Michaelis. *Michaelis - dicionário da brasileiro da língua portuguesa*, 2018.
- [47] George A. Miller. The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological Review*, 63(2):81–97, 1956.
- [48] Steven Minton. Quantitative results concerning the utility of explanation-based learning. In *Proceedings of AAAI-88*, volume 1, pages 564–569, St. Paul, Minnesota, 1988.
- [49] Tom M. Mitchell, Richard M. Keller, and Smadar T. Kedar-Cabelli. Explanation-based generalization: A unifying view. *Machine Learning*, 1(1):47–80, 1986.
- [50] Maria Candida Moraes. Informática educativa no brasil: uma história vivida, algumas lições aprendidas*. *Revista Brasileira de Informática na Educação*, 1:1–35, 1997.
- [51] Tom Murray. Authoring Intelligent Tutoring Systems: An analysis of the State of the Art. *International Journal of Artificial Intelligence*, 10(3):98–129, 1999.
- [52] Allen Newell and Herbert A. Simon. Human Problem Solving. *Contemporary Sociology*, 2(2):169, 1973.
- [53] Stephen C North and Murray Hill. Drawing graphs with dot 1 Basic Graph Drawing. *Small*, 1996.
- [54] Ludek Pachman. *Modern Chess Strategy*. Sir Isaac Pitman & Sons Ltd., London, 1963.
- [55] Marina Papastergiou. Digital Game-Based Learning in high school Computer Science education: Impact on educational effectiveness and student motivation. *Computers & Education*, 52(1):1–12, 2009.
- [56] Jacques Pitrat. A Chess Combination Program Which Uses Plan. *Artificial Intelligence*, 8:275–321, 1977.
- [57] Marc Prensky. *Digital Game Based Learning*. McGraw-Hill, New York, NY, USA, 2001.

- [58] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Series in Artificial Intelligence. Prentice Hall, Upper Saddle River, NJ, third edition, 2010.
- [59] Aleksander Sadikov and Ivan Bratko. Learning long-term chess strategies from databases. *Machine Learning*, 63(3):329–340, 2006.
- [60] Jonathan Schaeffer. Long-range planning in computer chess. In *Proceedings of the 1983 Annual Conference on Computers : Extending the Human Resource*, ACM '83, pages 170–179, New York, NY, EUA, 1983. ACM.
- [61] Reza Shams, Hermann Kaindl, and A Wien. Using Aspiration Windows for Minimax Algorithms. In *IJCAI Conference Proceedings*, pages 192–197, Sydney - Austrália, 1991. IJCAI.
- [62] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529:484–489, 2016.
- [63] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dhharshan Kumaran, Thore Graepel, Timothy P. Lillicrap, Karen Simonyan, and Demis Hassabis. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *CoRR*, abs/1712.01815, 2017.
- [64] Herbert a. Simon. The structure of ill structured problems. *Artificial Intelligence*, 4(3-4):181–201, 1973.
- [65] Herbert A Simon and Kevin Gilmarin. A simulation of memory for chess positions. *Cognitive Psychology*, 5(1):29 – 46, 1973.
- [66] David J. Slate and Lawrence R. Atkin. Chess 4.5 - The Northwestern University chess program. *Frey P.W (eds) Chess Skill in Man and Machine*, 1983.
- [67] S G Smith and B a Sherwood. Educational uses of the PLATO computer system. *Science (New York, N.Y.)*, 192:344–352, 1976.
- [68] P. Smolensky. Connectionist ai, symbolic ai, and the brain. *Artificial Intelligence Review*, 1(2):95–109, Jun 1987.
- [69] T. Soderstrom, L. Hall, T. Nilsson, and J. Ahlqvist. Computer Simulation Training in Health Care Education: Fuelling Reflection-in-Action? *Simulation & Gaming*, 45(6):805–828, 2015.
- [70] Andrew Soltis. *What it Takes to Become a Chess Master*. Pavilion Books, 2012.
- [71] Katharina Sonderegger-Iseli, Stefanie Burger, Jörg Muntwyler, and Franco Salomon. Diagnostic errors in three medical eras: a necropsy study. *The Lancet*, 355(9220):2027 – 2031, 2000.

- [72] Robert A. Sottolare, Arthur Graesser, Xiangen Hu, and Heather Holden. *Design Recommendations for Intelligent Tutoring Systems: Volume 1 - Learner Modeling*. US Army Research Laboratory, Orlando, Florida, 2013.
- [73] R Spiro, P Feltovich, Mj Jacobson, and Rl Coulson. Cognitive flexibility, Constructivism, and Hypertext: Random Access Instruction for Advanced Knowledge Aquisition in Ill-Structured Domains. *Educational Technology*, 315:24–33, 1992.
- [74] Kd Squire. Video games in education. *International Journal of Intelligent Simulations and Gaming*, 2:49—62, 2003.
- [75] David L. St-Pierre, Mark H. M. Winands, and David A. Watt. A selective move generator for the game axis and allies. In *Proceedings of 2010 IEEE Symposium on Computation Intelligence and Games (CIG)*, pages 162–169. IEEE, 2010.
- [76] Gerry Stahl, Timothy Koschmann, and Dan Suthers. Computer-supported collaborative learning: An historical perspective. *Cambridge handbook of the learning sciences*, pages 409–426, 2006.
- [77] Patrick Suppes. Current Trends in Computer-Assisted Instruction. *Advances in Computers. New York: Academic Press.*, 18(1972):173–229, 1979.
- [78] Prasad Tadepalli. Lazy explanation-based learning: A solution to the intractable theory problem. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence - Volume 1, IJCAI'89*, pages 694–700, San Francisco, CA, USA, 1989. Morgan Kaufmann Publishers Inc.
- [79] Gerhard Trippen. Plans, Patterns, and Move Categories Guiding a Highly Selective Search. *Advances in Computer Games*, pages 111–122, 2010.
- [80] Elisabet Tubau and Diego Alonso. Overcoming illusory inferences in a probabilistic counterintuitive problem: the role of explicit representations. *Memory & cognition*, 31(4):596–607, 2003.
- [81] J van Reek, Jwhm Uiterwijk, and H J van den Herik. Planning a strategy in chess. *Icca Journal*, 21:183–192, 1998.
- [82] Maria Virvou, Katsionis George, and Konstantinos Manos. Combining Software Games with Education: Evaluation of its Educational Effectiveness. *Educational Technology & Society*, 8(2):54–65, 2005.
- [83] John Von Neumann and O. Morgenstern. *The theory of games and economic behavior*. Princeton, 3 edition, 1990.
- [84] James L. Webster, William E. Reif, and Jeffrey S. Bracker. The Manager’s guide to strategic planning tools and techniques. *Strategy & Leadership*, 17(6):4–48, 1989.
- [85] Heinz Weihrich. The TOWS Matrix – A Tool for Situational Analysis. *Long Range Planning*, 15(2):54–66, 1982.
- [86] Etienne Wenger. *Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communications of Knowledge*. Morgan Kaufmann Publishers, Los Altos, California, 1987.

- [87] David Wilkins. Using patterns and plans in chess. In David Levy, editor, *Computer Chess Compendium*, pages 233–257. Springer New York, 1980.
- [88] Konstantin Zakharov, Antonija Mitrovic, and Lucy Johnston. Towards emotionally-intelligent pedagogical agents. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5091 LNCS:19–28, 2008.
- [89] Bj Zimmerman. Self-Efficacy: An Essential Motive to Learn. *Contemporary educational psychology*, 25(1):82–91, 2000.

Apêndice A: Definição da Linguagem de Padrões em Javacc

A listagem A.1 contém a definição completa da gramática da Linguagem de Especificação de Padrões utilizando Javacc.

```

1 options {
2   LOOKAHEAD = 1;
3   CHOICE_AMBIGUITY_CHECK = 2;
4   OTHER_AMBIGUITY_CHECK = 1;
5   STATIC = false;
6   DEBUG_PARSER = false;
7   DEBUG_LOOKAHEAD = false;
8   DEBUG_TOKEN_MANAGER = false;
9   ERROR_REPORTING = true;
10  JAVA_UNICODE_ESCAPE = false;
11  UNICODE_INPUT = false;
12  IGNORE_CASE = false;
13  USER_TOKEN_MANAGER = false;
14  USER_CHAR_STREAM = false;
15  BUILD_PARSER = true;
16  BUILD_TOKEN_MANAGER = true;
17  SANITY_CHECK = true;
18  FORCE_LA_CHECK = false;
19  OUTPUT_DIRECTORY = ".\\";
20 }
21
22
23 PARSER_BEGIN(LRPRGParser)
24
25 import java.util.HashMap;
26 import br.ufr.br.inf.heuchess.representacao.estrategia.ScenarioShape.PieceStatus;
27
28 public class LRPRGParser {
29
30   public class TS {
31     String Key;
32     int Category;
33     String Parameters = "";
34   }
35
36   public Pattern pattern;
37   public HashMap<String, TS> ST = new HashMap();
38   public String letfToken;
39   public ScenarioShape scenario; // elemento da tabela intermediaria
40
41   public static final int _FIGURE = 0;
42   public static final int _FUNCTION = 1;
43   private PieceStatus figStatus;
44
45   // as pos condicoes so podem existir se os movimentos previstos/esperados
46   // no padrao forem especificados ;
47
48   private boolean progMoves = false;
49
50   private ScenarioShape.PieceOwner owner;
51   private boolean ORclauseForm = false;
52
53   public static void main(String args[]) throws ParseException
54   {
55     try
56     {
57       LRPRGParser parser = new LRPRGParser(System.in);
58       parser.Patterns();
59       System.out.println("Tudo certo! :)");
60     }
61     catch (ParseException e)
62     {
63       System.out.println("Erro de sintaxe: " + e.getMessage());
64     }
65   }
66
67   // verifica se o pieceType passado na funcao movimentode sao do jogador que busca o padrao ou sao do oponente
68   public boolean myPiece(String pieceType)
69   {
70     return Character.isUpperCase(pieceType.charAt(0));
71   }
72 }
73
74
75 PARSER_END(LRPRGParser)
76

```

```

77 SKIP :
78 {
79   " "
80 | "\t"
81 | "\n"
82 | "\r"
83 }
84
85 /* Palavras reservadas */
86
87 TOKEN :
88 {
89   /* HLP */
90
91   <PATTERN : "PADRAO" | "PATTERN" > |
92   <AUTHOR : "AUTOR" | "AUTHOR" > |
93   <DESCRIPTION : "DESCRICA0" | "DESCRIPTION" > |
94   <WEIGHT : "PESO" | "WEIGHT" > |
95   <ACTORS : "ATOES" | "ACTORS" > |
96   <PIECESET : "SETOFPIECES" | "CONJUNTODEPECAS" > |
97   <FORMAT : "FORMAT" | "FORMATO" > |
98   <SCENARIO : "CENARIO" | "SCENARIO" > |
99   <MOVES : "TATICAS" | "TACTICALS" > |
100  <PRECONDITION : "PRECONDICAO" | "PRECONDITION" > |
101  <POSTCONDITION : "POSCONDICAO" | "POSTCONDITION" > |
102
103
104  /* BUILT-IN FUNCTIONS TO CHECK CONCEPTS AND CONDITIONS */
105
106  <WHITE : "WHITE" | "BRANCAS" > |
107  <BLACK : "BLACK" | "PRETAS" > |
108  <SQUARECOLOR : "CORCASADE" | "SQUARECOLOROF" > |
109  <ISINFILE : "EXISTINFILE" | "EXISTENACOLUNA" > |
110  <INSINRANK : "EXISTINRANK" | "EXISTENALINHA" > |
111  <FILE : "FILEOF" | "COLUNADE" > |
112  <RANK : "RANKOF" | "LINHADE" > |
113  <VALUE : "VALORDE" | "VALUEOF" > |
114  <ONEDGE : "NABORDA" | "ONEDGE" | "ONBORDER" > |
115  <ONCORNER : "NOCANTO" | "ONCORNER" > |
116  <TYPEOF : "TIPODE" | "TYPEOF" > |
117  <SENTRY : "SENTRY" | "SENTINELA" > |
118  <RAM : "RAM" | "BLOQUEADOR" > |
119  <MAXDISTANCE : "MAXDISTANCE" | "DISTANCIAMAX" > |
120  <MINDISTANCE : "MINDISTANCE" | "DISTANCIAMIN" > |
121  <RANKDISTANCE : "DISTANCIALIN" | "LINEDISTANCE" | "RANKDISTANCE" > |
122  <FILEDISTANCE : "COLDISTANCE" | "DISTANCIACOL" | "FILEDISTANCE" > |
123  <MANDISTANCE : "DISTANCIAMAN" | "MANHATTANDISTANCE" > |
124  <SQUARERULE : "NOQUADRADO" | "SQUARERULE" | "INTHESQUARE" > |
125
126  <MYDEFENSES : "MYDEFENSES" | "MINHASDEFESAS" > |
127  <YOURDEFENSES : "SUASDEFESAS" | "YOURDEFENSES" > |
128  <DEFENSES : "DEFESAS" | "DEFENSES" > |
129  <MYDEFENSESVALUE : "MYDEFENSESVALUE" | "VALORMINHASDEFESAS" > |
130  <YOURDEFENSESVALUE : "VALORSUASDEFESAS" | "YOURDEFENSESVALUE" > |
131  <DEFENSESVALUE : "VALORDEFESAS" | "DEFENSESVALUE" > |
132
133  <RELATIVEVALUEOF : "VALORRELATIVODE" | "RELATIVEVALUEOF" > |
134  <MYDIRDEFENSES : "MYDIRDEFENSES" | "MINHASDEFESASDIR" > |
135  <YOURDIRDEFENSES : "SUASDEFESASDIR" | "YOURDIRDEFENSES" > |
136  <DIRECTDEFENSES : "DEFESASDIRETAS" | "DIRECTDEFENSES" > |
137  <MYRELDEFENSESVALUE : "MYRELDEFENSESVALUE" | "VALORMINHASDEFESASREL" > |
138  <YOURRELDEFENSESVALUE : "VALORSUASDEFESASREL" | "YOURRELDEFENSESVALUE" > |
139  <RELDEFENSESVALUE : "VALORDEFESASREL" | "RELDEFENSESVALUE" > |
140  <MYDIRDEFENSESVALUE : "MYDIRDEFENSESVALUE" | "VALORMINHASDEFESASDIR" > |
141  <YOURDIRDEFENSESVALUE : "VALORSUASDEFESASDIR" | "YOURDIRDEFENSESVALUE" > |
142  <DIRDEFENSESVALUE : "VALORDEFESASDIR" | "DIRECTDEFENSESVALUE" > |
143  <WEDGES : "INTRUSOSENTE" | "WEDGESBETWEEN" > |
144
145  /* retorna o numero de movimentos possiveis da peca atuando no padrao ou de um tipo especificado */
146  <MOVESOF : "MOVESOF" | "MOVIMENTOSDE" > |
147  /* retorna o numero de movimentos seguros possiveis da peca atuando no padrao ou de um tipo especificado */
148  <SAFEMOVESOF : "SAFEMOVESOF" | "MOVIMENTOSSEGUROSDE" > |
149  // retorna um valor inteiro > 0 se existe um caminho seguro do rei ate a peca/quadrado
150  <KINGPATH : "KINGPATHTO" | "CAMINHODOREI" > |
151  // retorna verdadeiro se existe uma instancia de um outro padrao na posicao
152  <CHECKPATTERN : "EXISTEPADRAO" | "EXISTPATTERN" > |
153  // determina se os reis estao em perfeita oposicao e retorna logico (verdadeiro ou falso)
154  <KINGOPPOSITION : "OPOSICAO" | "KINGOPPOSITION" > |
155  // retorna verdadeiro se nenhum peao do jogador especificado como parametro pode se mover
156  <BLOCKEDPAWNS : "BLOCKEDPAWNS" | "PEOSBLOQUEADOS" > |
157
158  /* ACTORS */
159
160  <KING : "K" | "k" > |
161  <QUEEN : "Q" | "q" > |
162  <ROOK : "R" | "r" > |
163  <BISHOP : "B" | "b" > |
164  <KNIGHT : "N" | "n" > |
165  <PAWN : "P" | "p" > |
166  <OWNER : "EU" | "ME" | "OPONENTE" | "OPONENT" > |
167  <FOLLOWNER : <OWNER> | "NONE" | "VAZIA" | "ANYONE" | "NINGUEM" > |
168  <WILDCARD : "F" | "f" | "S" | "s" > |
169  <OP_AND : "E" | "AND" > |
170  <OP_OR : "OU" | "OR" > |
171  <FALSE : "FALSO" | "FALSE" > |
172  <TRUE : "TRUE" | "VERDADEIRO" > |
173  <DIMENSION : ["a"-"h"] > |
174
175  /* ACTIONS */

```

```

176
177 <OP_ATTACK : "=>" | "ATTACKS" | "ATAACA" > |
178 <OP_IATTACK : "->" | "INDIRECTATTACK" | "ATACAINDIRETO" > |
179 <OP_DEFENSE : "<" | "DEFENDS" | "DEFENDE" > |
180 <OP_IDEFENSE : "<-" | "INDIRECTDEFENSE" | "DEFESAINDIRETA" > |
181 <OP_MOVE : ">>" | "MOVE" | "MOVES" > |
182 <OP_BLOCK : "||" | "BLOCKS" | "BLOCKS" > |
183 <OP_RIGHTSIDE : "|>" | "RIGHTOF" | "DIREITADE" > |
184 <OP_LEFTSIDE : "<|" | "LEFTOF" | "ESQUERDADE" > |
185 <OP_TOPSIDE : "|^" | "ABOVEOF" | "UPPERSIDE" > |
186 <OP_BOTTOMSIDE : "|v" | "UNDEROF" | "UNDERSIDE" > |
187 <OP_SE : "\\>" | "SUDESTE" | "SOUTHEAST" > |
188 <OP_NW : "<\\|" | "NOROESTE" | "NORTHWEST" > |
189 <OP_NE : "/>" | "NORDESTE" | "NORTHEAST" > |
190 <OP_SW : "</" | "SUDOESTE" | "SOUTHWEST" > |
191 <OP_PATTACK : ":>" | "POSSIBLEATTACK" | "ATAQUEPOSSIVEL" > |
192 <OP_PDEFENSE : ";<" | "DEFESAPOSSIVEL" | "POSSIBLEDEFENSE" > |
193 <OP_CHECKMATE : "#>" | "CHECKMATE" | "CHECKMATE" > |
194 <OP_NOT : "NAO" | "NOT" > |
195
196 <#SIGNAL : "-" | "+" > |
197 <UINT : ["0"- "9"] (["0"- "9"])* > |
198 <REAL : (<SIGNAL>)?(["0"- "9"])+ "." (["0"- "9"])* > |
199 <TEXT :
200     "\n"
201     (
202         (~[\"\"'\"\\\" \r"])
203         | (\"\\\"
204           ( [\"n\", \"t\", \"b\", \"r\", \"f\", \"\\\", \"/\", \"'\"]
205             | [\"0\"- \"7\"] ( [\"0\"- \"7\"] )?
206               | [\"0\"- \"3\"] [\"0\"- \"7\"] [\"0\"- \"7\"]
207             )
208         )
209     )*
210     "\n" >
211 }
212 /* IDENTIFIERS */
213
214 /** Root production. */
215 void Patterns() :
216 {}
217 {
218     (Pattern())*<EOF>
219 }
220
221 void Pattern() :
222 {Token t;
223  String s;}
224 {
225     <PATTERN> t = <TEXT> { pattern.setNome(t.image.substring(1, t.image.length()-1)); } /* objeto pattern deve ser inicializado
226     antes de acionar o parser e definido na variavel pattern */
227     <AUTHOR> <TEXT>
228     <DESCRIPTION> t = <TEXT> { pattern.setDescription(t.image); }
229     <WEIGHT> t = <REAL> { pattern.setWeight(Double.parseDouble(t.image)); }
230     (<ACTORS> Atores())?
231     (<PIECESET> ExclusiveSet())?
232     (LOOKAHEAD(2) <FORMAT> | <SCENARIO>) ScenarioShapePattern()
233     { StringBuilder sb = new StringBuilder();
234       for(ScenarioShape f:pattern.getScenarioMembers()) {
235         f.rightFigSource.addAll(f.rightFig);
236         sb.append(f.toString()).append("\n");
237       }
238       pattern.setScenario(sb.toString());
239     }
240     (<PRECONDITION> s = LogicExpression() {pattern.setPreConditionToJava(s);})?
241     (<MOVES> Moves() { progMoves = true; })?
242     (<POSTCONDITION> s = LogicExpression()
243      { if (!progMoves) {
244        throw new ParseException("Pos-condicoes exigem sequencia de movimentos previstos/esperados.");
245      }
246      pattern.setPosConditionToJava(s);
247     })?
248     ";"
249 }
250
251 void Relation() :
252 {}
253 {
254     (<OP_ATTACK> {scenario.relationType = PatternRelationType.ATTACKS; } |
255     <OP_DEFENSE> {scenario.relationType = PatternRelationType.DEFENDS; } |
256     <OP_IATTACK> {scenario.relationType = PatternRelationType.INDIRECTATTACK; } |
257     <OP_IDEFENSE> {scenario.relationType = PatternRelationType.INDIRECTDEFENSE; } |
258     <OP_MOVE> {scenario.relationType = PatternRelationType.MOVE; } |
259     <OP_BLOCK> {scenario.relationType = PatternRelationType.BLOCKS; } |
260     <OP_TOPSIDE> {scenario.relationType = PatternRelationType.UPPERSIDE; } |
261     <OP_RIGHTSIDE> {scenario.relationType = PatternRelationType.RIGHTSIDE; } |
262     <OP_LEFTSIDE> {scenario.relationType = PatternRelationType.LEFTSIDE; } |
263     <OP_BOTTOMSIDE> {scenario.relationType = PatternRelationType.UNDERSIDE; } |
264     <OP_SE> {scenario.relationType = PatternRelationType.SE; } |
265     <OP_NW> {scenario.relationType = PatternRelationType.NW; } |
266     <OP_NE> {scenario.relationType = PatternRelationType.NE; } |
267     <OP_SW> {scenario.relationType = PatternRelationType.SW; } |
268     <OP_PATTACK> {scenario.relationType = PatternRelationType.POSSIBLEATTACK; } |
269     <OP_PDEFENSE> {scenario.relationType = PatternRelationType.POSSIBLEDEFENSE; } |
270     <OP_CHECKMATE> {scenario.relationType = PatternRelationType.CHECKMATE; } )
271 }
272
273 void Atores() :

```

```

274 {String t;}
275 {
276     t = figura0()
277     {
278         TS simb = ST.get(t);
279         if(simb == null) {
280             simb = new TS();
281             simb.Key = t;
282             simb.Category = _FIGURE;
283             ST.put(simb.Key, simb);
284         }
285         scenario = new ScenarioShape();
286         scenario.orIni = false;
287         scenario.orFim = false;
288         scenario.leftFig = t;
289         scenario.leftStatus = figStatus;
290         scenario.relationType = PatternRelationType.DECLARATION;
291         if("FfSs".indexOf(t.substring(0,1)) >= 0) {
292             throw new ParseException("Coringas nao devem ser declarados como atores");
293         }
294         if(Character.isUpperCase(t.charAt(0)))
295         {
296             scenario.leftOwner = ScenarioShape.PieceOwner.MINE;
297         } else if(Character.isLowerCase(t.charAt(0)))
298         {
299             scenario.leftOwner = ScenarioShape.PieceOwner.YOURS;
300         }
301         pattern.getScenarioMembers().add(scenario);
302     }
303     (",")
304     t = figura0()
305     {
306         simb = ST.get(t);
307         if(simb == null) {
308             simb = new TS();
309             simb.Key = t;
310             simb.Category = _FIGURE;
311             ST.put(simb.Key, simb);
312         }
313         scenario = new ScenarioShape();
314         scenario.orIni = false;
315         scenario.orFim = false;
316         scenario.leftFig = t;
317         scenario.leftStatus = figStatus;
318         scenario.relationType = PatternRelationType.DECLARATION;
319         if("FfSs".indexOf(t.substring(0,1)) >= 0) {
320             throw new ParseException("Coringas nao devem ser declarados como atores");
321         }
322         if(Character.isUpperCase(t.charAt(0)))
323         {
324             scenario.leftOwner = ScenarioShape.PieceOwner.MINE;
325         } else if(Character.isLowerCase(t.charAt(0)))
326         {
327             scenario.leftOwner = ScenarioShape.PieceOwner.YOURS;
328         }
329         pattern.getScenarioMembers().add(scenario);
330     }
331     )*
332 }
333
334 void ScenarioDescription() :
335 {String t;}
336 {
337     t = Figure1()
338     {
339         TS simb = ST.get(t);
340         if(simb == null) {
341             simb = new TS();
342             simb.Key = t;
343             simb.Category = _FIGURE;
344             simb.Parameters = ORClauseForm?"OR":"";
345             ST.put(simb.Key, simb);
346         }
347         scenario = new ScenarioShape();
348         scenario.orIni = false;
349         scenario.orFim = false;
350         scenario.leftFig = t;
351         scenario.leftStatus = figStatus;
352         if(t.charAt(0) == 's' || t.charAt(0) == 's')
353         {
354             throw new ParseException("Referencia coringa <" + t + "> nao pode estar isolada ou ao lado esquerdo da relacao: \n"
355                                     + "Linha: " + token.beginLine());
356         } else if(Character.isUpperCase(t.charAt(0)))
357         {
358             scenario.leftOwner = ScenarioShape.PieceOwner.MINE;
359         } else if(Character.isLowerCase(t.charAt(0)))
360         {
361             scenario.leftOwner = ScenarioShape.PieceOwner.YOURS;
362         }
363     }
364     ( Relation() {scenario.notRelation = false;} | <OP_NOT> Relation() {scenario.notRelation = true;} )
365     Figure2()
366     {
367         pattern.getScenarioMembers().add(scenario);
368     }
369 }
370
371 void ScenarioShapePattern() :
372 {Token t;}

```

```

373 {
374 (
375 ScenarioDescription()
376 |
377 " {"
378 { ORClauseForm = true; }
379 (ScenarioDescription() { scenario.orIni = true; })+
380 "}"
381 {scenario.orFim = true; ORClauseForm = false;}
382 )+
383 }
384
385 String PieceType() :
386 {Token t;}
387 {
388 t = <KING> { return myPiece(t.image)?"K":"k"; }
389 t = <QUEEN> { return myPiece(t.image)?"Q":"q"; }
390 t = <BISHOP> { return myPiece(t.image)?"B":"b"; }
391 t = <KNIGHT> { return myPiece(t.image)?"N":"n"; }
392 t = <ROOK> { return myPiece(t.image)?"R":"r"; }
393 t = <PAWN> { return myPiece(t.image)?"P":"p"; }
394 }
395
396 String Piece() :
397 {Token t;
398 String s;}
399 {
400 t = <WILDCARD> { return t.image; } | s = PieceType() { return s; }
401 }
402
403 String figura0() :
404 { Token t1;
405 String s;
406 }
407 {
408 s = PieceType() t1 = <UINT> { figStatus = PieceStatus.ANYONE; return s + t1.image; }
409 }
410
411 String Figure1() :
412 { Token t1;
413 String s;
414 }
415 {
416 LOOKAHEAD(2)
417 s = Piece() t1 = <UINT> { figStatus = PieceStatus.ANYONE; return s + t1.image; }
418 |
419 LOOKAHEAD(2)
420 "<" s = Piece() t1 = <UINT> ">" { figStatus = PieceStatus.PROTECTED; return s + t1.image; }
421 |
422 LOOKAHEAD(2)
423 ">" s = Piece() t1 = <UINT> "<" { figStatus = PieceStatus.UNPROTECTED; return s + t1.image; }
424 }
425
426 void Figure2() :
427 { Token t;
428 String s;
429 TS simb;
430 }
431 { //relacoes diretas
432 s = Figure1()
433 {
434 if(s.charAt(0) == 's' || s.charAt(0) == 's') {
435 owner = ScenarioShape.PieceOwner.ANYONE;
436 } else if (Character.isUpperCase(s.charAt(0)))
437 {
438 owner = ScenarioShape.PieceOwner.MINE;
439 } else
440 {
441 owner = ScenarioShape.PieceOwner.YOURS;
442 }
443 if(scenario.relationType == PatternRelationType.ATTACKS && scenario.leftOwner==owner) {
444 throw new ParseException("Pecas do mesmo jogador nao podem atacar-se mutuamente." +
445 "\nLinha: " + token.beginLine);
446 } else if(scenario.relationType == PatternRelationType.ATTACKS && owner == ScenarioShape.PieceOwner.ANYONE) {
447 throw new ParseException("Uma peca nao pode atacar uma casa coringa, somente defender." +
448 "\nLinha: " + token.beginLine);
449 } else if(scenario.relationType == PatternRelationType.DEFENDS && owner!=scenario.leftOwner && owner != ScenarioShape.
450 PieceOwner.ANYONE) {
451 throw new ParseException("Pecas de um jogador nao podem defender pecas do outro."
452 + "\nLinha: " + token.beginLine);
453 } else if(scenario.relationType != PatternRelationType.MOVE &&
454 scenario.relationType != PatternRelationType.DEFENDS &&
455 scenario.relationType != PatternRelationType.CHECKMATE &&
456 owner==ScenarioShape.PieceOwner.ANYONE) {
457 throw new ParseException("Operador relacional invalido. Utilize o operador de movimento ou defesas direta."
458 + "\nLinha: " + token.beginLine);
459 } else if(!scenario.relationType.getDireta()) {
460 {if (true) throw new ParseException("Operador relacional invalido. Utilize operadores diretos."
461 + "\u005cnLinha: " + token.beginLine);}
462 } else {
463 boolean jaexiste = false;
464 for(int ix = 0; ix < scenario.rightFig.size(); ix++) {
465 if (s.equals(scenario.rightFig.get(ix))) {
466 jaexiste = true; break;
467 }
468 }
469 if(s.equals(scenario.leftFig) || jaexiste) {
470 throw new ParseException("Uma figura nao pode interagir com ela mesma!");
471 }

```

```

471     }
472     simb = ST.get(s);
473     if (simb == null) {
474         simb = new TS();
475         simb.Key = s;
476         simb.Category = _FIGURE;
477         simb.Parameters = ORClauseForm?"OR":"";
478         ST.put(simb.Key, simb);
479     }
480     scenario.rightFig.add(s);
481     scenario.rightStatus.add(figStatus);
482     //redefine o owner para operacoes de consulta
483     if (Character.isUpperCase(s.charAt(0)))
484     {
485         owner = ScenarioShape.PieceOwner.MINE;
486     } else
487     {
488         owner = ScenarioShape.PieceOwner.YOURS;
489     }
490     scenario.rightOwner.add(owner);
491 }
492 | //relacoes indiretas
493 " ("
494 s = Figure1()
495 {
496     simb = ST.get(s);
497     if (simb == null) {
498         simb = new TS();
499         simb.Key = s;
500         simb.Category = _FIGURE;
501         simb.Parameters = ORClauseForm?"OR":"";
502         ST.put(simb.Key, simb);
503     }
504     scenario.rightFig.add(s);
505     scenario.rightStatus.add(figStatus);
506     if (Character.isUpperCase(s.charAt(0))) {
507         owner = ScenarioShape.PieceOwner.MINE;
508     } else {
509         owner = ScenarioShape.PieceOwner.YOURS;
510     }
511     scenario.rightOwner.add(owner);
512 }
513 " ("
514 s = Figure1()
515 {
516     simb = ST.get(s);
517     if (simb == null) {
518         simb = new TS();
519         simb.Key = s;
520         simb.Category = _FIGURE;
521         simb.Parameters = ORClauseForm?"OR":"";
522         ST.put(simb.Key, simb);
523     }
524     scenario.rightFig.add(s);
525     scenario.rightStatus.add(figStatus);
526     if (s.charAt(0) == 's' || s.charAt(0) == 's') {
527         owner = ScenarioShape.PieceOwner.ANYONE;
528     } else
529     if (Character.isUpperCase(s.charAt(0))) {
530         owner = ScenarioShape.PieceOwner.MINE;
531     } else {
532         owner = ScenarioShape.PieceOwner.YOURS;
533     }
534     scenario.rightOwner.add(owner);
535 })+
536 {
537     if ((scenario.relationType == PatternRelationType.INDIRECTATTACK ||
538         scenario.relationType == PatternRelationType.POSSIBLEATTACK ||
539         scenario.relationType == PatternRelationType.ATTACKS) &&
540         (owner == scenario.leftOwner || owner == ScenarioShape.PieceOwner.ANYONE)) {
541         throw new ParseException("Pecas do mesmo jogador nao podem atacar-se mutuamente ou coringas S."
542             + "\nLinha: " + token.beginLine);
543     } else if ((scenario.relationType == PatternRelationType.INDIRECTDEFENSE ||
544         scenario.relationType == PatternRelationType.POSSIBLEDEFENSE) &&
545         owner != scenario.leftOwner &&
546         owner != ScenarioShape.PieceOwner.ANYONE) {
547         throw new ParseException("O dono da ultima peca da lista deve ser o mesmo da peca a esquerda ou a peca deve ser um
548             coringa S."
549             + "\nLinha: " + token.beginLine);
550     } else if (scenario.relationType.getDireta()) {
551         if (true) throw new ParseException("Operador relacional inv\u00e9lido. Utilize operadores indiretos."
552             + "\u005cnLinha: " + token.beginLine);
553     }
554 }
555 }
556
557 void Moves() :
558 {String arg1 = "", arg2 = "";
559 }
560 {
561     (
562     arg1 = Figure1()
563     {
564         TS simb = ST.get(arg1);
565         if (simb == null) {
566             throw new ParseException("A figura [" + arg1 + "] deve estar declarada no scenario/atores do padrao.");
567         }
568     }

```

```

569     ", "
570     arg2 = Figure1()
571     {
572         simb = ST.get(arg2);
573         if(simb == null) {
574             throw new ParseException("A figura [" + arg2 + "] deve estar declarada no cenário/atores do padrão.");
575         }
576         if(arg2.equals(arg1)) {
577             throw new ParseException("O destino do movimento deve ser diferente da origem");
578         }
579         if(Character.toUpperCase(arg1.charAt(0)) == 'S') {
580             throw new ParseException("O primeiro elemento do movimento deve ser uma peça do conjunto [FKQRNBP].");
581         }
582         if(Character.toUpperCase(arg2.charAt(0)) != 'S')
583             if((Character.toUpperCase(arg1.charAt(0)) &&&
584                 Character.toUpperCase(arg2.charAt(0))) ||
585                 (Character.toLowerCase(arg1.charAt(0)) &&&
586                 Character.toLowerCase(arg2.charAt(0)))) {
587                 throw new ParseException("Movimento sobre peça própria e inválido!");
588             }
589         pattern.getMoves().add(arg1 + "," + arg2); //adiciona a tabela IL de movimentos
590     }
591 )+
592 }
593
594 String LogicExpression():
595 {String s1 = "",s2 = "",s3 = "";}
596 {
597     s1 = ANDExpression() ( <OP_OR> {s2 = " || "; } s3 = ANDExpression() {s1 += s2 + s3; } ) *
598     { return s1; }
599 }
600
601 String ANDExpression():
602 {String s = "", s1 = "",s2 = "",s3 = "";}
603 {
604     s1 = EqualExpression() { s = s1; pattern.getPreConditions().add(s1); }
605     ( <OP_AND> { s2 = " && "; } s3 = EqualExpression()
606     {
607         pattern.getPreConditions().add(s3);
608         s += s2 + s3;
609     } ) *
610     { return s; }
611 }
612
613 String EqualExpression():
614 {String s1 = "",s3 = "";
615  Token s2 = null;
616 }
617 {
618     s1 = RelationalExpression() ( ( s2 = "=" | s2 = "!=" ) s3 = RelationalExpression()
619     {s1 += s2.image=="?"+" + s3:s2.image + s3; } ) *
620     { return s1; }
621 }
622
623 String RelationalExpression():
624 {String s1 = "",s3 = "";
625  Token s2 = null;
626 }
627 {
628     s1 = AddExpression() ( ( s2 = "<" | s2 = ">" | s2 = "<=" | s2 = ">=" ) s3 = AddExpression()
629     { s1 += s2.image + s3; } ) *
630     { return s1; }
631 }
632
633 String AddExpression():
634 {String s1 = "", s3 = "";
635  Token s2 = null;
636 }
637 {
638     s1 = MulExpression() ( ( s2 = "+" | s2 = "-" ) s3 = MulExpression()
639     { s1 += s2.image + s3; } ) *
640     { return s1; }
641 }
642
643 String MulExpression():
644 {String s1 = "",s3 = "";
645  Token s2 = null;
646 }
647 {
648     s1 = UnaryExpression() ( ( s2 = "*" | s2 = "/" | s2 = "%" ) s3 = UnaryExpression()
649     { s1 += s2.image + s3; } ) *
650     {
651         return s1;
652     }
653 }
654
655 String UnaryExpression():
656 { Token t1 = null;
657  Token t = null;
658  String s1;
659 }
660 {
661     ( t1 = "+" | t1 = "-" ) ? ( t = <UINT> | t = <REAL> )
662     { if(t1==null) {return t.image; } else { return t1.image + t.image; } }
663     | t = <TRUE>
664     { return "true"; }
665     | t = <FALSE>
666     { return "false"; }
667 }

```

```

668     s1 = BuiltInFunction()
669     {return s1; }
670 |
671 | (" s1 = LogicExpression() ")
672 | { return "(" + s1 + ")"; }
673 | }
674 |
675 | /* Funcoes pre-definidas */
676 |
677 | String BuiltInFunction():
678 | { String f = "", args = "";
679 |   Token arg;}
680 |
681 | //-----
682 |
683 | { // zero parametros
684 |   LOOKAHEAD(2)
685 |   ( <WHITE>           { f = "analise.WHITEPIECES()"; } // bool
686 |   | <BLACK>          { f = "analise.BLACKPIECES()"; } // bool
687 |   )
688 | }
689 | //-----
690 |
691 | | // um parametro FIGURA
692 | | ( <FILE>           { f = "analise.FILEOF"; } // INTEIRO_SEM_SINAL
693 | | | <RANK>           { f = "analise.RANKOF"; } // INTEIRO_SEM_SINAL
694 | | | <ONEDGE>        { f = "analise.ONEDGE"; } // BOOL
695 | | | <ONCORNER>      { f = "analise.ONCORNER"; } // BOOL
696 | | | <MYDEFENSES>    { f = "analise.MYDEFENSES"; } // INTEIRO_SEM_SINAL
697 | | | <YOURDEFENSES>  { f = "analise.YOURDEFENSES"; } // INTEIRO_SEM_SINAL
698 | | | <MYDEFENSESVALUE> { f = "analise.MYDEFENSESVALUE"; } // REAL
699 | | | <YOURDEFENSESVALUE> { f = "analise.YOURDEFENSESVALUE"; } // REAL
700 | | /* valores relativos de defesas conforme a posicao de origem do defensor */
701 | | | <RELATIVEVALUEOF> { f = "analise.RELATIVEVALUEOF"; } // REAL
702 | | | <MYDIRDEFENSES>   { f = "analise.MYDIRDEFENSES"; } // INTEIRO_SEM_SINAL
703 | | | <YOURDIRDEFENSES> { f = "analise.YOURDIRDEFENSES"; } // INTEIRO_SEM_SINAL
704 | | | <MYRELDEFENSESVALUE> { f = "analise.MYRELDEFENSESVALUE"; } // REAL
705 | | | <YOURRELDEFENSESVALUE> { f = "analise.YOURRELDEFENSESVALUE"; } // REAL
706 | | | <MYDIRDEFENSESVALUE> { f = "analise.MYDIRDEFENSESVALUE"; } // REAL
707 | | | <YOURDIRDEFENSESVALUE> { f = "analise.YOURDIRDEFENSESVALUE"; } // REAL
708 | | | <SENTRY>         { f = "analise.SENTRY"; } // INTEIRO_SEM_SINAL
709 | | | <RAM>            { f = "analise.RAM"; } // INTEIRO_SEM_SINAL
710 | | ) (" args = Args() ")
711 | | { return f + "(" + args + ")"; }
712 | | }
713 | | //-----
714 | |
715 | | | //um parametro - dono/lado da busca/verificacao
716 | | | <BLOCKEDPAWNS> (" arg = <OWNER> ")
717 | | | { f = "analise.ISPAWNSBLOCKED(\"" + arg.image + "\")";
718 | | |   return f;
719 | | | } // bool true se todos os peoes nao conseguem avançar
720 | | | }
721 | | //-----
722 | |
723 | | | //um parametro : ator ou tipo da peca
724 | | | <SQUARECOLOR> { f = "analise.SQUARECOLOROF"; }
725 | | | | (" (LOOKAHEAD(2) args = Args() | args = Piece()) ") //INTEIRO_SEM_SINAL
726 | | | | { return f + "(" + args + ")"; }
727 | | | | }
728 | | | |
729 | | | | <VALUE> { f = "analise.VALUEOF"; } // REAL
730 | | | | | (" (LOOKAHEAD(2) args = Args() | args = PieceType()) ")
731 | | | | | { return f + "(" + args + ")"; }
732 | | | | | }
733 | | | | |
734 | | | | | <MOVESOF> { f = "analise.MOVESOF"; }
735 | | | | | | (" (LOOKAHEAD(2) args = Args() | args = PieceType()) ") //INTEIRO_SEM_SINAL
736 | | | | | | { return f + "(" + args + ")"; }
737 | | | | | | }
738 | | | | | |
739 | | | | | | <SAFEMOVESOF> { f = "analise.SAFEMOVESOF"; }
740 | | | | | | | (" (LOOKAHEAD(2) args = Args() | args = PieceType()) ") // INTEIRO_SEM_SINAL
741 | | | | | | | { return f + "(" + args + ")"; }
742 | | | | | | | }
743 | | | | | | //-----
744 | | | | | |
745 | | | | | | //dois parametros FIGURA e DONO
746 | | | | | | (
747 | | | | | | | <DEFENSES> { f = "analise.DEFENSESOF"; } |
748 | | | | | | | <DEFENSESVALUE> { f = "analise.DEFENSESVALUEOF"; } | |
749 | | | | | | | <DIRECTDEFENSES> { f = "analise.DIRECTDEFENSESOF"; } | |
750 | | | | | | | <DIRECTDEFENSESVALUE> { f = "analise.DIRECTDEFENSESVALUEOF"; } | |
751 | | | | | | | <RELDDEFENSESVALUE> { f = "analise.RELDEFENSESVALUEOF"; }
752 | | | | | | | )
753 | | | | | | | args = OwnerArgs()
754 | | | | | | | {return f + args; }
755 | | | | | | | }
756 | | | | | | |
757 | | | | | | | //dois parametros FIGURA e FULLOWNER
758 | | | | | | | <SQUAREOWNER> { f = "analise.SQUAREOWNER"; }
759 | | | | | | | args = SquareOwnerArgs()
760 | | | | | | | { return f + args; }
761 | | | | | | | }
762 | | | | | | | //-----
763 | | | | | | |
764 | | | | | | | //dois parametros FIGURA e TIPO
765 | | | | | | | <TYPEOF> { f = "analise.TYPEOF"; }
766 | | | | | | | args = TypeArgs() // BOOL
767 | | | | | | | { return f + args; }
768 | | | | | | | }
769 | | | | | | |
770 | | | | | | | <ISINFILE> { f = "analise.EXISTINFILE"; }
771 | | | | | | | args = FileExistArgs()
772 | | | | | | | }

```

```

767     { return f + args; } // BOOL
768 |
769 | <INSINRANK> { f = "analise.EXISTINRANK"; }
770 args = RankExistArgs()
771 { return f + args; } // BOOL
772 |
773 | ( <KINGPATH> { f = "analise.KINGPATHTO"; } // BOOL
774 | <KINGOPPOSITION> { f = "analise.KINGOPPOSITION"; } // bool
775 | <RANKDISTANCE> { f = "analise.RANKDISTANCE"; } // INTEIRO_SEM_SINAL
776 | <FILEDISTANCE> { f = "analise.FILEDISTANCE"; }
777 | <MANDISTANCE> { f = "analise.MANHATTANDISTANCE"; }
778 | <MAXDISTANCE> { f = "analise.MAXDISTANCE"; }
779 | <MINDISTANCE> { f = "analise.MINDISTANCE"; }
780 | <SQUARERULE> { f = "analise.INTHESQUARE"; } // BOOL
781 | <WEDGES> { f = "analise.WEDGESBETWEEN"; }
782 )
783 args = DistanceArgs()
784 { return f + args; }
785 | <CHECKPATTERN> "(" args = PatternArgs() ")"
786 { f = "analise.EXISTPATTERN(\"" + args + "\")";
787   return f;
788 }
789 }
790
791 String ExclusiveSet():
792 {String t = "", s = " ";}
793 {
794   (s = PieceType() { pattern.getExclusiveSet().add(s.charAt(0)); t = t + s; } )+
795   {
796     if(pattern.getExclusiveSet().size() < 3) {
797       throw new ParseException("Tres pecas no minimo devem compor o conjunto exclusivo.");
798     }
799     return t;
800   }
801 }
802 }
803
804 String PatternArgs():
805 { Token t;
806   Token d;
807 }
808 {
809   t = <TEXT>
810   ", "
811   d = <OWNER>
812   {
813     pattern.setRecursiveSearch(d.image.startsWith("OP"));
814     return t.image.substring(1,t.image.length()-1) + ", " + d.image;
815   }
816 }
817
818 String Args():
819 { String s; }
820 {
821   s = Figure1()
822   {
823     TS simb = ST.get(s);
824     if(simb == null)
825     {
826       throw new ParseException("A figura [" + s + "] deve estar declarada no scenario do padrao.");
827     }
828     return s;
829   }
830 }
831
832 String OwnerArgs():
833 {String arg1 = "", arg2 = "";
834   Token t = null;}
835 {
836   "("
837   arg1 = Figure1()
838   ", "
839   t = <OWNER> { arg2 = t.image; }
840   ")"
841   { return "\" + arg1 + ", " + arg2 + "\""; }
842 }
843
844 String SquareOwnerArgs():
845 {String arg1 = "", arg2 = "";
846   Token t = null;}
847 {
848   "("
849   arg1 = Figure1()
850   ", "
851   t = <FULLOWNER> { arg2 = t.image; }
852   ")"
853   { return "\" + arg1 + ", " + arg2 + "\""; }
854 }
855
856 String FileExistArgs():
857 {String arg1 = "", arg2 = "", arg3 = "";
858   Token t = null;}
859 {
860   "("
861   arg1 = Piece()
862   ", "
863   (
864     t = <UINT>
865     { arg2 = t.image; }

```

```

866 |         |
867 |         <FILE> "(" arg3 = Args() ")"
868 |         { arg2 = "analise.COLUNADE(" + arg3 + ")"; }
869 |     )
870 |     "}"
871 |     { return "(" + arg1 + "," + arg2 + "\)"; }
872 | }
873 |
874 | String RankExistArgs():
875 | {String arg1 = "", arg2 = "", arg3 = "";
876 |   Token t = null;}
877 | {
878 |   "("
879 |   arg1 = Piece()
880 |   ","
881 |   ( t = <UINT>
882 |   { arg2 = t.image; }
883 |   |
884 |   <RANK> "(" arg3 = Args() ")"
885 |   { arg2 = "analise.COLUNADE(" + arg3 + ")"; }
886 |   )
887 |   "}"
888 |   {return "(" + arg1 + "," + arg2 + "\)"; }
889 | }
890 |
891 | String TypeArgs():
892 | {String arg1 = "", arg2 = "";}
893 | {
894 |   "("
895 |   arg1 = Figure1()
896 |   {
897 |     TS simb = ST.get(arg1);
898 |     if(simb == null) {
899 |       throw new ParseException("A figura [" + arg1 + "] deve estar declarada no cenario do padrao.");
900 |     }
901 |   }
902 |   ","
903 |   arg2 = PieceType()
904 |   {
905 |     simb = ST.get(arg2);
906 |     if(simb == null) {
907 |       simb = new TS();
908 |       simb.Key = arg2;
909 |       simb.Category = _FUNCTION;
910 |       ST.put(simb.Key, simb);
911 |     }
912 |   }
913 |   "}"
914 |   {return "(" + arg1 + "," + arg2 + "\)"; }
915 | }
916 |
917 |
918 | String DistanceArgs():
919 | {String arg1 = "", arg2 = "";}
920 | {
921 |   "("
922 |   arg1 = Figure1()
923 |   {
924 |     TS simb = ST.get(arg1);
925 |     if(simb == null) {
926 |       throw new ParseException("A figura [" + arg1 + "] deve estar declarada no cenario do padrao.");
927 |     }
928 |   }
929 |   ","
930 |   arg2 = Figure1()
931 |   {
932 |     simb = ST.get(arg2);
933 |     if(simb == null) {
934 |       throw new ParseException("A figura [" + arg2 + "] deve estar declarada no cenario do padrao.");
935 |     }
936 |   }
937 |   "}"
938 |   {return "(" + arg1 + "," + arg2 + "\)"; }
939 | }

```

Listagem A.1: Gramática da Linguagem PATLAN em Javacc

Apêndice B: Tutorial da Linguagem PATLAN

TUTORIAL DE USO DA LINGUAGEM DE REPRESENTAÇÃO DE PADRÕES (PATLAN).

Autor: Luis C. F. Bueno – Curitiba – Setembro/2017

1 Introdução

A Linguagem de Representação de Padrões (PATLAN – do inglês Pattern Language) é uma linguagem desenvolvida para uso pela comunidade enxadrística que permite a representação e/ou especificação de padrões que ocorrem em partidas de xadrez.

Entende-se por padrão qualquer subconjunto de peças que estejam atuando em uma posição de jogo, através de posicionamentos táticos e relacionamentos de ataque/defesa/controle(movimento)/bloqueio ou posicionamento relativo.

Opcionalmente, pode ter um conjunto de conceitos verificáveis (condições iniciais), um conjunto de movimentos táticos de execução do padrão, e um conjunto de conceitos finais necessários para que o padrão seja validado. A correta representação deste subconjunto pode resultar em posições estratégicas dentro de um plano de jogo.

Alguns exemplos de padrões, considerados estratégicos, são os já conhecidos padrões de mate, entre eles: Philidor, Cozio, Anastasia. Tais padrões podem ocorrer em posições bem diferentes de jogo.

A linguagem foi incorporada ao Ambiente StratVision, e possui um módulo de interpretação e busca do padrão em posições de jogo, durante uma partida ou em bases de jogos já executados, a fim de estimular o estudo/aprendizado sobre padrões. Também apresenta a característica de ser bilíngue. Os termos e palavras-chaves podem ser traduzidos para seu equivalente em inglês, com exceção da notação algébrica de peças que é somente a internacional (língua inglesa).

2 Descrevendo Padrões com a PATLAN

A PATLAN é dividida em 8 seções distintas:

1. Documentação
2. Valoração heurística
3. Atores (opcional)
4. Conjunto exclusivo (para representar tipos específicos de finais – opcional)
5. Cenário inicial (peças e relacionamentos iniciais)
6. Condições iniciais (conceitos de alto nível verificáveis através de funções pré-definidas da PATLAN – opcional)
7. Movimentos táticos – sequência opcional de movimentos programados para o padrão

8. Condições finais – opcional, conceitos de alto nível finais a serem verificados na posição final.

2.1 Visão Geral

A listagem abaixo apresenta o modelo de estruturação da linguagem:

1. PADRAO "Nome do padrão"
2. AUTOR "Nome do autor"
3. DESCRICAO "Texto descritivo sobre o padrão"
4. PESO <valor de importância heurística do padrão>
5. [ATORES <lista de peças participantes>]
6. [CONJUNTODEPECAS <lista de tipos de peças>]
7. CENARIO <especificação do cenário>
8. [PRECONDICAO <condições lógicas iniciais>]
9. [TATICAS <sequência de movimentos programados>]
10. [POSCONDICAO <condições finais do padrão>]

As duas primeiras linhas referem-se a parte documental do padrão.

A linha 1 obrigatoriamente deve começar com a cláusula PADRAO seguido de um nome entre aspas.

A linha 2 deve começar com a cláusula DESCRICAO seguido de uma descrição sobre o padrão entre aspas. Esta descrição pode ser qualquer informação útil que explique o padrão especificado.

A palavra PESO é usada para definir um valor heurístico dado pelo autor do padrão que estabeleça o grau de importância do padrão. Tal peso será utilizado como fator multiplicador do valor heurístico de um tabuleiro numa competição automática, podendo ser positivo se a estratégia é conduzir o jogo para o mesmo, ou negativo, se o objetivo é desviar dele.

A cláusula ATORES existe para relacionar as peças participantes do cenário buscado, mas que não necessariamente possuem um relacionamento com outra peça ou quadrado. Serve para denotar a simples existência da peça na posição de jogo.

A cláusula CONJUNTODEPECAS permite designar tipos específicos e exclusivos que devem existir na posição de jogo. Ideal para indicar a busca de padrões em tipos de finais específicos, exemplo: KNk, para finais de Rei e Cavalo x Rei.

2.2 Cláusula FORMATO / CENARIO

A cláusula FORMATO identifica a seção do código que descreve de forma genérica os elementos que estão atuando no padrão. Considerando que padrões ocorrem do ponto de vista de ambos os lados da partida, e a fim de unificar a representação, de tal forma que uma mesma representação possa ser usado para ambos os lados, adotou-se a seguinte convenção, inspirado no modelo algébrico de representação enxadrístico, com a seguinte variação: peças representadas em caixa alta (maiúsculas) pertencem ao lado que está buscando o padrão e peças representadas em caixa baixa (minúscula) referem-se às do oponente.

Cada linha da cláusula formato designa uma relação posicional ou semântica entre duas ou mais peças (dependendo do tipo de relação), da seguinte forma:

<Peça><Identificador numérico> <relação> <Peça><Identificador numérico>

	K1 ATACA r1
	K1 ATACA r2
Ou:	
	K1 => r1
	K1 => r2

Quadro 1 - Exemplo de formato

O Quadro 1 acima descreve um padrão onde o Rei (branco ou preto, dependendo do lado que seja deseja buscar) ataca simultaneamente duas torres adversárias (um garfo produzido pelo Rei). Assim, por exemplo, as duas posições representadas na Figura 1 estão contempladas na representação.

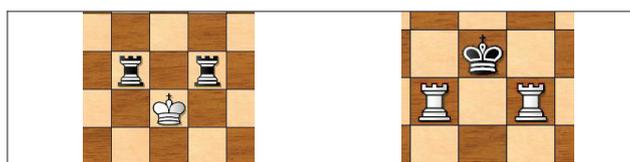


Figura 1 - Garfo duplo (Rei x Torres)

Toda referência a um tipo de peça deve ser seguido de um identificador numérico. Isto é necessário devido à duplicidade das peças de longo alcance e da multiplicidade de peões, além da possibilidade de coroamentos durante o jogo. A única peça que não há possibilidade de duplicação é o Rei, mas por questões de padronização, deve ser designado um identificador numérico, mesmo que esta seja a peça representada.

2.2.1 Relações Posicionais

Uma representação é dita posicional quando ela descreve posicionamentos de peças em relação de uma com outra. Exemplo:

	P1 DIREITADE P2
	P1 ACIMADE P3
Ou:	
	P1 < P2
	P1 ^ P3

Quadro 2 Relações posicionais - exemplo

O Quadro 2 representa a seguinte situação, lembrando que esta mesma representação poderia ser aplicada para peças pretas.



Figura 2 Relações posicionais - exemplo

Veja a lista completa de relações posicionais no final deste tutorial.

2.2.2 Relações Semânticas

São definidas em termos das regras de operação das peças entre si em termos de ataques e defesas. Estas relações são divididas em duas categorias: (a) relações diretas/estáticas e (b) relações indiretas/dinâmicas. As relações diretas são os fatos que estão acontecendo na posição e as relações indiretas, são os fatos que podem acontecer se um movimento for realizado pela peça protagonista da ação representada. Veja alguns exemplos de relações diretas no Quadro 3.

	P1 ATACA q1
	P2 DEFENDE P1
	R1 DEFENDE P2
Ou:	
	P1 => q1
	P2 =< P1
	R1 =< P2

Quadro 3 Relações semânticas diretas - exemplo

O cenário descrito acima representa qualquer posição de jogo em que isto seja verdadeiro, independente da posição de R1 (veja figura 3).

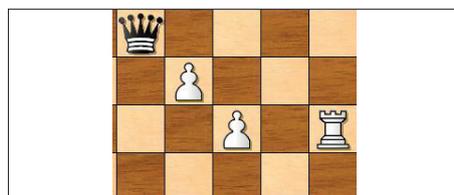


Figura 3 Relações semânticas diretas - exemplo

As relações indiretas ou dinâmicas, dizem respeito a uma situação de ataque ou defesa que podem ocorrer se a peça protagonista (à esquerda da relação) realizar um movimento válido. Veja no Quadro 4 alguns exemplos de como representar tais cenários.

	Q1 ATACAINDIRETO (S1,k1)
	R1 DEFENDEINDIRETO (S1,K1)
	N1 ATAQUEPOSSIVEL (p1,r1)
	B1 ATAQUEPOSSIVEL (S1,b1)
	B1 DEFESAPOSSIVEL (S1, Q1)
Ou:	

Q1 -> (S1,k1)
R1 -< (S1,K1)
N1 -> (p1,r1)
B1 :-> (S1,b1)
B1 :< (S1, Q1)

Quadro 4 Relações semânticas indiretas - exemplo

A primeira linha representa uma situação onde a Dama de um jogador realiza um movimento qualquer para uma casa vazia ou ocupada pelo adversário e ataca o Rei adversário colocando-o em xeque. A segunda linha indica uma situação onde uma Torre pode se mover para uma casa e defender o seu Rei. A terceira linha estabelece uma situação onde um Cavalo através de uma captura de um peão adversário, ameaça/ataca uma Torre adversária. A quarta linha indica um movimento possível de ser realizado por um Bispo, ameaçando um bispo do adversário, que necessariamente esteja em outra diagonal. A última linha descreve uma situação onde o Bispo pode se mover para uma casa e dali defender a Dama que está em outra diagonal.

2.3 Cláusula CONDICAÇÃO/POSCONDICAÇÃO

Estas cláusulas permitem explicitar condições específicas e restritivas sobre o padrão representado na cláusula formato. A condição é definida na forma de acionamento de funções de verificação sobre as peças definidas no formato, e sempre resulta numa avaliação lógica de retorno verdadeiro ou falso.

O Quadro 5 apresenta um exemplo de como validar se duas peças F1 e F2 estejam na mesma coluna.

CONDICAÇÃO
COLUNADE(F1) = COLUNADE(F2);
POSCONDICAÇÃO
MOVIMENTOSDE(k1) = 0

3 Lista de Operações da Cláusula FORMATO/CENARIO

Operação	Descrição	Símbolo alternativo
RELAÇÕES SEMÂNTICAS		
ATACA	Ataque direto.	F1 => f1
DEFENDE	Defesa direta.	F1 =< F2
ATACAINDIRETO	Ataque indireto onde a peça atacada está no raio de ação da peça representada à esquerda da operação, mas tem uma outra peça em seu caminho. Exemplo na mesma diagonal de um Bispo, ou na mesma linha ou coluna de uma Torre.	F1 -> (f1, f2) F1 -> (F2,f1)
DEFENDEINDIRETO	Defesa indireta onde a peça defendida está no raio de ação da peça representada à esquerda da operação, mas tem uma outra peça em seu caminho. Exemplo na mesma diagonal de um Bispo, ou na mesma linha ou coluna de uma Torre.	F1 -< (f1, F2) F1 -< (F2,F3)
MOVE	Indica um movimento direto válido.	F1 >> S1
ATAQUEPOSSIVEL	Indica um ataque a uma peça através de um movimento, cujo local está fora do raio de ação de movimento da peça que ataca.	F1 :> (S1,f1) F1 :> (f1, f2) F1 :> (F2,f1)
DEFESAPOSSIVEL	Indica uma defesa de uma peça através de um movimento, cujo local está fora do raio de ação de movimento da peça que defende.	F1 :< (S1,F2) F1 :< (f1, F2) F1 :< (F2,F3)
RELAÇÕES POSICIONAIS – relativas à cor da peça (acima é o lado do oponente)		
ACIMADE	Uma peça na casa de cima da outra.	F1 ^ F2
ABAIXODE	Uma peça na casa de baixo da outra.	F1 v F2
DIREIDADE	Uma peça à direita de outra.	F1 < F2
ESQUERDADE	Uma peça à esquerda de outra.	F1 > F2
NORDESTE	Uma peça à nordeste de outra.	F1 /> F2
SUDESTE	Uma peça à sudeste de outra.	F1 \> F2
NOROESTE	Uma peça à noroeste de outra.	F1 <\ F2
SUDOESTE	Uma peça à sudoeste de outra.	F1 </ F2

4 Lista de Funções das Cláusulas CONDICAÇÃO/POSCONDIÇÃO

Função	Descrição	Parâmetros
FUNÇÕES COM APENAS UM PARÂMETRO		
COLUNADE	Retorna o número da coluna (1 a 8).	Ator/figura do formato: COLUNADE(F1) = 2 COLUNADE(B1) = 8
CORCASA	Retorna 0 se a casa é branca, 1 se for preta.	Ator/figura do formato: CORCASA(F1) = 1 CORCASA(B1) = 0
LINHADE	Retorna o número da linha (1 a 8).	Ator/figura do formato: LINHADE(F1) = 2 LINHADE(B1) = 8
MOVIMENTOSDE	Retorna o número de movimentos válidos possíveis do ator passado como parâmetro	Ator/figura do formato: MOVIMENTOSDE(K1)=0
NABORDA	Retorna Verdadeiro se o parâmetro está em uma das bordas do tabuleiro, falso caso contrário.	Ator/figura do formato: NABORDA(K1)
NOCANTO	Retorna Verdadeiro se o parâmetro está em um dos cantos do tabuleiro, falso caso contrário	Ator/figura do formato: NOCANTO(K1)
SENTINELA	Retorna Verdadeiro se existe algum peão adversário atuando como sentinela de um peão especificado no formato.	Ator/figura do formato: SENTINELA(P1)
VALORDE	Retorna o valor heurístico do ator. Inicialmente com a seguinte valoração: Peão = 1 Torre = 2 Cavalo = 3 Bispo = 3 Dama = 9.	Ator/figura do formato: VALORDE(F1) > VALORDE(F2)
MINHASDEFESAS ¹	Retorna quantas peças do jogador estão defendendo a casa/ator.	MINHASDEFESAS(F1) MINHASDEFESAS(S1)
SUASDEFESAS ¹	Retorna quantas peças do oponente estão defendendo a casa/ator.	SUASDEFESAS(F1) SUASDEFESAS(S1)
VALORMINHASDEFESAS ¹	Retorna o somatório dos valores das peças que defendem uma casa/ator definida no padrão.	VALORMINHASDEFESAS(F1) VALORMINHASDEFESAS(S1)
VALORSUASDEFESAS ¹	Retorna o somatório dos valores das peças do oponente que defendem uma posição definida no padrão.	VALORSUASDEFESAS(F1) VALORSUASDEFESAS(S1)
FUNÇÕES COM DOIS PARÂMETROS		
DISTANCIACOL	Retorna a distância em colunas entre os parâmetros	Primeiro parâmetro: ator Segundo parâmetro: ator DISTANCIACOL(F1,f1) = 3
DISTANCIALIN	Retorna a distância em linhas entre os parâmetros	Primeiro parâmetro: ator Segundo parâmetro: ator DISTANCIALIN(F1,f1) = 3
DISTANCIAMAN	Retorna a distância Manhattan	Primeiro parâmetro: ator

¹ Todas as peças que defendem diretamente uma posição são computadas no somatório + a soma de todas as peças que defendem indiretamente essa posição através das peças que as defendem diretamente.

	entre os parâmetros (linhas + colunas)	Segundo parâmetro: ator DISTANCIAMAN(F1,f1)
DISTANCIAMAX	Retorna a maior distância entre os parâmetros (em linhas ou colunas)	Primeiro parâmetro: ator Segundo parâmetro: ator DISTANCIAMAX(F1,f1) = 3
DISTANCIAMIN	Retorna a menor distância entre os parâmetros (em linhas ou colunas)	Primeiro parâmetro: ator Segundo parâmetro: ator DISTANCIAMIN(F1,f1) = 3
TIPODE	Retorna verdadeiro se o tipo do ator do primeiro parâmetro é igual ao tipo especificado no segundo parâmetro.	Primeiro parâmetro: ator Segundo parâmetro: tipo da peça. TIPODE(F1,DAMA) OU TIPODE(F1,TORRE)
DEFESAS	Retorna quantas peças do jogador indicado estão defendendo uma casa ou peça de forma direta e indireta.	Primeiro parâmetro: ator Segundo parâmetro: lado DEFESAS(f1,OPONENTE) DEFESAS(F1,EU)
DEFESASDIR	Retorna quantas peças do jogador indicado estão defendendo uma casa ou peça de forma direta.	Primeiro parâmetro: ator Segundo parâmetro: lado DEFESASDIR(f1,OPONENTE) DEFESASDIR(F1,EU)
VALORDEFESAS	Retorna o somatório dos valores das peças defendendo uma peça/casa direta e indiretamente, de acordo com o lado informado.	Primeiro parâmetro: ator Segundo parâmetro: lado VALORDEFESAS(F1,OPONENTE) VALORDEFESAS(F1,EU)
VALORDEFESASDIR	Retorna o somatório dos valores das peças defendendo uma peça/casa diretamente, de acordo com o lado informado.	Primeiro parâmetro: ator Segundo parâmetro: lado VALORDEFESASDIR(f1,OPONENTE) VALORDEFESASDIR(F1,EU)
VALORDEFESASREL	Retorna o valor relativo agindo sobre uma casa*	
DONODACASA	Retorna verdadeiro se o dono é igual ao segundo parâmetro	Primeiro parâmetro: ator Segundo parâmetro: dono que pode ser: EU, Oponente, VAZIA, NINGUEM DONODACASA(S1,NINGUEM)

* O cálculo é baseado na seguinte fórmula:

Valor heurístico do ator x ((col + lin x TOTAL LINHAS) + 1).

Isto permite identificar de qual tipo de peça/ator parte a defesa, e de que parte do tabuleiro vem a defesa (col,lin).

5 Cláusula MOVIMENTOS / TATICAS

Essa cláusula permite indicar a sequência de movimentos exigida pelo padrão. Cada movimento é representado em uma linha do texto, com duas peças participantes do padrão separados por vírgula. Exemplo:

MOVIMENTOS
F1,S2
f1,S2
N1,S1

EXEMPLOS DE PADRÕES REPRESENTADOS

```

PADRAO "GARFODUPLO"
  AUTOR "Luis Bueno"
  DESCRICAO "<F1> ataca <f2> e <f3>"
  PESO 0.8 //medida de forca do padrao
  CENARIO
    F1 ATACA f2
    F1 ATACA f3;

PADRAO "GARFODINAMICO"
  AUTOR "Luis Bueno"
  DESCRICAO "<F1> move para <S1> de onde ataca <f2> e <f3>"
  PESO 0.8
  CENARIO
    F1 ATACA (S1,f2)
    F1 ATACA (S1,f3);

PADRAO "ATAQUEDESCOBERTO"
  AUTOR "Luis Bueno"
  DESCRICAO "<F1> ataca <f3> indiretamente por <F2>"
  PESO 0.6
  CENARIO
    F1 ATACA (F2,f3)
  PRECONDICAO
    VALORDE(F1) > VALORDE(P) AND
    VALORDE(F1) != VALORDE(N);

    // OU TIPODE(F1,P) = FALSO AND TIPODE(F1,N) = FALSO

PADRAO "RAIOX"
  AUTOR "Luis Bueno"
  DESCRICAO "<F1> ataca <f3> indiretamente por <F2>"
  PESO 0.6
  CENARIO
    F1 ATACA (f2,f3)
  PRECONDICAO
    (TIPODE(F1,Q) OU
    TIPODE(F1,R) OU
    TIPODE(F1,B)) E
    VALORDE(f3) < VALORDE(f2);

PATTERN "COZIO"
  AUTHOR "Luis Bueno"
  DESCRIPTION "Cozio in 3 plies"

```

```
WEIGHT 999
SCENARIO
  Q1 POSSIBLEATTACK (>s1<,k1)
  k1 DEFENDS p1
  k1 DEFENDS s2
  p1 DEFENDS f1
  K1 POSSIBLEATTACK (s3,f1)
  K1 MOVE >s4<
PRECONDITION
  TYPEOF(f1,N) = FALSE AND
  (FILEOF(p1) = FILEOF(k1) OR
  RANKOF(p1) = RANKOF(k1)) AND
  (FILEOF(s1) = FILEOF(s4) OR
  RANKOF(s1) = RANKOF(s4)) AND
  ONBORDER(k1) AND
  (COLDISTANCE(s4,k1) = 2 OR
  LINEDISTANCE(s4,k1) = 2) AND
  (FILEOF(s3) = FILEOF(f1) OR
  RANKOF(s3) = RANKOF(f1)) AND
  (FILEOF(s2) = FILEOF(p1) OR
  RANKOF(s2) = RANKOF(p1)) AND
  (FILEOF(s2) = FILEOF(f1) OR
  RANKOF(s2) = RANKOF(f1))
TACTICALS
  Q1,s1
  k1,s2
  s1,s4
POSTCONDITION
  MOVESOF(k1)= 0;
```

6 Cláusula MOVIMENTOS / TATICAS

Esta seção apresenta as palavras-chaves em português e seu equivalente em inglês:

Português	Inglês
PADRAO	PATTERN
AUTOR	AUTHOR
DESCRICA0	DESCRIPTION
ATORES	ACTORS
CONJUNTODEPECAS	SETOFPIECES
PESO	WEIGHT
CENARIO	SCENARIO
PRECONDICAO	PRECONDITION
TATICAS/MOVIMENTOS	TACTICALS
Ações	
ABAIXODE	UNDERSIDE
ACIMADE	UPPERSIDE
ATACA	ATTACKS
ATAQUEINDIRETO	INDIRECTATTACK
ATAQUEPOSSIVEL	POSSIBLEATTACK
BLOQUEIA	BLOCKS
DEFENDE	DEFENDS
DEFESAINDIRETA	INDIRECTDEFENSE
DEFESAPOSSIVEL	POSSIBLEDEFENSE
DIREITADE	RIGHTOF
DIREITADE	LEFTOF
NAO	NOT
NORDESTE	NORTHEAST
NOROESTE	NORTHWEST
SUDESTE	SOUTHEAST
SUDOESTE	SOUTHWEST
XEQUEMATE	CHECKMATE
Funções condicionais/conceitos	
BLOQUEADOR	RAM
BRANCAS	WHITE
CAMINHODOREI	KINGPATHTO
COLUNADE	FILEOF
CORCASADE	SQUARECOLOROF
DEFESAS	DEFENSES
DISTANCIACOL	FILEDISTANCE
DISTANCIALIN	RANKDISTANCE
DISTANCIAMAX	MAXDISTANCE
DISTANCIAMIN	MINDISTANCE
DONODACASA	SQUAREOWNER

EU	ME
EXISTENACOLUNA	EXISTINFILE
EXISTENALINHA	EXISTINRANK
EXISTEPADRAO	EXISTPATTERN
FALSO	FALSE
INTRUSOSENTE	WEDGESBETWEEN
LINHADE	RANKOF
MINHASDEFESDIR	MYDIRDEFENSES
MINHASDEFESAS	MYDEFENSES
MOVIMENTOSDE	MOVESOF
MOVIMENTOSSEGUROSDE	SAFEMOVESOF
MYDEFENSESVALUE	VALORMINHASDEFESAS
NABORDA	ONEDGE/ONBORDER
NINGUEM	NONE
NOCANTO	ONCORNER
NOQUADRADO	SQUARERULE
OPONENTE	OPPONENT
OPOSICAO	KINGOPPOSITION
PECASDEF	PIECESDEF
PEOESBLOQUEADOS	BLOCKEDPAWNS
PRETAS	BLACK
SENTINELA	SENTRY
SUASDEFESAS	YOURDEFENSES
SUASDEFESASDIR	YOURDIRDEFENSES
TIPODE	TYPEOF
VALOMINHASDEFESASREL	MYRELDEFENSESVALUE
VALORDE	VALUEOF
VALORDEFESAS	DEFENSESVALUE
VALORDEFESASDIR	DIRECTDEFENSESVALUE
VALORDEFESASREL	YOURRELDEFENSESVALUE
VALORRELATIVODE	RELATIVEVALUEOF
VALORSUASDEFESAS	YOURDEFENSESVALUE
VALORSUASDEFESASDIR	YOURDIRDEFENSESVALUE
VALORSUASDEFESASREL	YOURRELDEFENSESVALUE
VAZIA	ANYONE
VERDADEIRO	TRUE

Apêndice C: Exemplo de Base de Fatos Prolog

A listagem abaixo contém a base de fatos gerada para a posição exemplificada na figura 4.5.

```

1 ataca(mBb3,spa4).
2 ataca(mBb3,spe6).
3 ataca(mPe5,snf6).
4 ataca(mPf5,spe6).
5 ataca(mQe2,spb5).
6 ataca(null,null).
7 ataca(spa4,mBb3).
8 ataca(spe6,mPf5).
9 ataqueindireto(mBb3,ce6,skf7).
10 ataqueindireto(mBb3,spe6,skf7).
11 ataqueindireto(mQe2,ce5,spe6).
12 ataqueindireto(mQe2,mPe5,spe6).
13 ataqueindireto(mRf1,cf5,snf6).
14 ataqueindireto(mRf1,mPf5,snf6).
15 ataqueindireto(null,null,null).
16 ataqueindireto(sbd7,ce6,mPf5).
17 ataqueindireto(sbd7,spe6,mPf5).
18 ataqueindireto(sqc8,ce6,mPc3).
19 ataqueindireto(sqc8,spc6,mPc3).
20 ataqueindireto(srd8,cd7,mBd6).
21 ataqueindireto(srd8,sbd7,mBd6).
22 ataqueindireto(sre8,ce6,mPe5).
23 ataqueindireto(sre8,spe6,mPe5).
24 ataquepossivel(mBb3,ca4,spb5).
25 ataquepossivel(mBb3,cc4,spb5).
26 ataquepossivel(mBb3,cd5,spc6).
27 ataquepossivel(mBb3,ce6,sbd7).
28 ataquepossivel(mBb3,spa4,spb5).
29 ataquepossivel(mBb3,spe6,sbd7).
30 ataquepossivel(mBd6,cc7,srd8).
31 ataquepossivel(mBd6,ce5,snf6).
32 ataquepossivel(mBd6,ce7,snf6).
33 ataquepossivel(mBd6,ce7,srd8).
34 ataquepossivel(mBd6,cf8,spg7).
35 ataquepossivel(mBd6,mPe5,snf6).
36 ataquepossivel(mPc3,cc4,spb5).
37 ataquepossivel(mPd4,cd5,spc6).
38 ataquepossivel(mPd4,cd5,spe6).
39 ataquepossivel(mPe5,cf6,spg7).
40 ataquepossivel(mPe5,snf6,spg7).
41 ataquepossivel(mPf5,ce6,sbd7).
42 ataquepossivel(mPf5,ce6,skf7).
43 ataquepossivel(mPf5,spe6,sbd7).
44 ataquepossivel(mPf5,spe6,skf7).
45 ataquepossivel(mQe2,ca2,spa4).
46 ataquepossivel(mQe2,cb5,spa4).
47 ataquepossivel(mQe2,cb5,spc6).
48 ataquepossivel(mQe2,cc4,spa4).
49 ataquepossivel(mQe2,cc4,spc6).
50 ataquepossivel(mQe2,cc4,spe6).
51 ataquepossivel(mQe2,cd2,sph6).
52 ataquepossivel(mQe2,ce3,sph6).
53 ataquepossivel(mQe2,ce4,spc6).
54 ataquepossivel(mQe2,ce5,snf6).
55 ataquepossivel(mQe2,cf3,spc6).
56 ataquepossivel(mQe2,cg2,spc6).
57 ataquepossivel(mQe2,cg2,spg7).
58 ataquepossivel(mQe2,cg4,spg7).
59 ataquepossivel(mQe2,ch5,skf7).
60 ataquepossivel(mQe2,ch5,sph6).
61 ataquepossivel(mQe2,mPa2,spa4).
62 ataquepossivel(mQe2,mPe5,snf6).
63 ataquepossivel(mQe2,mPg2,spc6).
64 ataquepossivel(mQe2,mPg2,spg7).
65 ataquepossivel(mQe2,spb5,spa4).
66 ataquepossivel(mQe2,spb5,spc6).
67 ataquepossivel(null,null,null).
68 ataquepossivel(sbd7,cc6,mPg2).
69 ataquepossivel(sbd7,ce6,mBb3).
70 ataquepossivel(sbd7,spc6,mPg2).
71 ataquepossivel(sbd7,spe6,mBb3).
72 ataquepossivel(snf6,cd5,mPc3).
73 ataquepossivel(snf6,cd7,mPe5).
74 ataquepossivel(snf6,ce4,mBd6).
75 ataquepossivel(snf6,ce4,mPc3).
76 ataquepossivel(snf6,ce8,mBd6).
77 ataquepossivel(snf6,cg4,mPe5).
78 ataquepossivel(snf6,cg4,mPh2).

```

79 ataquepossivel(snf6,sbd7,mPe5).
80 ataquepossivel(snf6,sre8,mBd6).
81 ataquepossivel(sng8,ce7,mPf5).
82 ataquepossivel(sng8,ch6,mPf5).
83 ataquepossivel(sng8,sph6,mPf5).
84 ataquepossivel(spa4,cb3,mPa2).
85 ataquepossivel(spa4,mBb3,mPa2).
86 ataquepossivel(spb5,cb4,mPc3).
87 ataquepossivel(spc6,cc5,mPd4).
88 ataquepossivel(spg7,cc6,mPf5).
89 ataquepossivel(sqc8,cb8,mBd6).
90 ataquepossivel(sqc8,cc6,mBd6).
91 ataquepossivel(sqc8,cc6,mPg2).
92 ataquepossivel(sqc8,cc7,mBd6).
93 ataquepossivel(sqc8,cd7,mBd6).
94 ataquepossivel(sqc8,sbd7,mBd6).
95 ataquepossivel(sqc8,spc6,mBd6).
96 ataquepossivel(sqc8,spc6,mPg2).
97 ataquepossivel(sre8,ce6,mBd6).
98 ataquepossivel(sre8,spe6,mBd6).
99 b(mBb3).
100 b(mBd6).
101 b(null).
102 b(sbd7).
103 casa(ca1).
104 casa(ca2).
105 casa(ca3).
106 casa(ca4).
107 casa(ca5).
108 casa(ca6).
109 casa(ca7).
110 casa(ca8).
111 casa(cb1).
112 casa(cb2).
113 casa(cb3).
114 casa(cb4).
115 casa(cb5).
116 casa(cb6).
117 casa(cb7).
118 casa(cb8).
119 casa(cc1).
120 casa(cc2).
121 casa(cc3).
122 casa(cc4).
123 casa(cc5).
124 casa(cc6).
125 casa(cc7).
126 casa(cc8).
127 casa(cd1).
128 casa(cd2).
129 casa(cd3).
130 casa(cd4).
131 casa(cd5).
132 casa(cd6).
133 casa(cd7).
134 casa(cd8).
135 casa(ce1).
136 casa(ce2).
137 casa(ce3).
138 casa(ce4).
139 casa(ce5).
140 casa(ce6).
141 casa(ce7).
142 casa(ce8).
143 casa(cf1).
144 casa(cf2).
145 casa(cf3).
146 casa(cf4).
147 casa(cf5).
148 casa(cf6).
149 casa(cf7).
150 casa(cf8).
151 casa(cg1).
152 casa(cg2).
153 casa(cg3).
154 casa(cg4).
155 casa(cg5).
156 casa(cg6).
157 casa(cg7).
158 casa(cg8).
159 casa(ch1).
160 casa(ch2).
161 casa(ch3).
162 casa(ch4).
163 casa(ch5).
164 casa(ch6).
165 casa(ch7).
166 casa(ch8).
167 defende(mBb3,ca2).
168 defende(mBb3,ca4).
169 defende(mBb3,cc2).
170 defende(mBb3,cc4).
171 defende(mBb3,cd1).
172 defende(mBb3,cd5).
173 defende(mBb3,ce6).
174 defende(mBb3,mPa2).
175 defende(mBd6,ca3).
176 defende(mBd6,cb4).
177 defende(mBd6,cb8).

178 defende(mBd6,cc5).
179 defende(mBd6,cc7).
180 defende(mBd6,ce5).
181 defende(mBd6,ce7).
182 defende(mBd6,cf8).
183 defende(mBd6,mPe5).
184 defende(mKg1,cf1).
185 defende(mKg1,cf2).
186 defende(mKg1,cg2).
187 defende(mKg1,ch1).
188 defende(mKg1,ch2).
189 defende(mKg1,mPg2).
190 defende(mKg1,mPh2).
191 defende(mKg1,mRf1).
192 defende(mPa2,cb3).
193 defende(mPa2,mBb3).
194 defende(mPc3,cb4).
195 defende(mPc3,cd4).
196 defende(mPc3,mPd4).
197 defende(mPd4,cc5).
198 defende(mPd4,ce5).
199 defende(mPd4,mPe5).
200 defende(mPe5,cd6).
201 defende(mPe5,cf6).
202 defende(mPe5,mBd6).
203 defende(mPf5,ce6).
204 defende(mPf5,cg6).
205 defende(mPg2,cf3).
206 defende(mPg2,ch3).
207 defende(mPh2,cg3).
208 defende(mQe2,ca2).
209 defende(mQe2,cb2).
210 defende(mQe2,cb5).
211 defende(mQe2,cc2).
212 defende(mQe2,cc4).
213 defende(mQe2,cd1).
214 defende(mQe2,cd2).
215 defende(mQe2,cd3).
216 defende(mQe2,ce1).
217 defende(mQe2,ce3).
218 defende(mQe2,ce4).
219 defende(mQe2,ce5).
220 defende(mQe2,cf1).
221 defende(mQe2,cf2).
222 defende(mQe2,cf3).
223 defende(mQe2,cg2).
224 defende(mQe2,cg4).
225 defende(mQe2,ch5).
226 defende(mQe2,mPa2).
227 defende(mQe2,mPe5).
228 defende(mQe2,mPg2).
229 defende(mQe2,mRe1).
230 defende(mQe2,mRf1).
231 defende(mRe1,ca1).
232 defende(mRe1,cb1).
233 defende(mRe1,cc1).
234 defende(mRe1,cd1).
235 defende(mRe1,ce2).
236 defende(mRe1,cf1).
237 defende(mRe1,mQe2).
238 defende(mRe1,mRf1).
239 defende(mRf1,ce1).
240 defende(mRf1,cf2).
241 defende(mRf1,cf3).
242 defende(mRf1,cf4).
243 defende(mRf1,cf5).
244 defende(mRf1,cg1).
245 defende(mRf1,mKg1).
246 defende(mRf1,mPf5).
247 defende(mRf1,mRe1).
248 defende(null, null).
249 defende(sbd7,cc6).
250 defende(sbd7,cc8).
251 defende(sbd7,ce6).
252 defende(sbd7,ce8).
253 defende(sbd7,spc6).
254 defende(sbd7,spe6).
255 defende(sbd7,sqc8).
256 defende(sbd7,sre8).
257 defende(skf7,ce6).
258 defende(skf7,ce7).
259 defende(skf7,ce8).
260 defende(skf7,cf6).
261 defende(skf7,cf8).
262 defende(skf7,cg6).
263 defende(skf7,cg7).
264 defende(skf7,cg8).
265 defende(skf7,snf6).
266 defende(skf7,sng8).
267 defende(skf7,spe6).
268 defende(skf7,spg7).
269 defende(skf7,sre8).
270 defende(snf6,cd5).
271 defende(snf6,cd7).
272 defende(snf6,ce4).
273 defende(snf6,ce8).
274 defende(snf6,cg4).
275 defende(snf6,cg8).
276 defende(snf6,ch5).

277 defende(snf6,ch7).
278 defende(snf6,sbd7).
279 defende(snf6,sng8).
280 defende(snf6,sre8).
281 defende(sng8,ce7).
282 defende(sng8,cf6).
283 defende(sng8,ch6).
284 defende(sng8,snf6).
285 defende(sng8,sph6).
286 defende(spa4,cb3).
287 defende(spb5,ca4).
288 defende(spb5,cc4).
289 defende(spb5,spa4).
290 defende(spc6,cb5).
291 defende(spc6,cd5).
292 defende(spc6,spb5).
293 defende(spe6,cd5).
294 defende(spe6,cf5).
295 defende(spg7,cf6).
296 defende(spg7,ch6).
297 defende(spg7,snf6).
298 defende(spg7,sph6).
299 defende(sph6,cg5).
300 defende(sqc8,ca6).
301 defende(sqc8,ca8).
302 defende(sqc8,cb7).
303 defende(sqc8,cb8).
304 defende(sqc8,cc6).
305 defende(sqc8,cc7).
306 defende(sqc8,cd7).
307 defende(sqc8,cd8).
308 defende(sqc8,sbd7).
309 defende(sqc8,spc6).
310 defende(sqc8,srd8).
311 defende(srd8,cc8).
312 defende(srd8,cd7).
313 defende(srd8,ce8).
314 defende(srd8,sbd7).
315 defende(srd8,sqc8).
316 defende(srd8,sre8).
317 defende(sre8,cd8).
318 defende(sre8,ce6).
319 defende(sre8,ce7).
320 defende(sre8,cf8).
321 defende(sre8,cg8).
322 defende(sre8,sng8).
323 defende(sre8,spe6).
324 defende(sre8,srd8).
325 defesaindireta (mBb3,ce6,ca2).
326 defesaindireta (mBb3,ce6,mPa2).
327 defesaindireta (mBb3,spe6,ca2).
328 defesaindireta (mBb3,spe6,mPa2).
329 defesaindireta (mBd6,ce5,cf4).
330 defesaindireta (mBd6,ce5,cg3).
331 defesaindireta (mBd6,ce5,ch2).
332 defesaindireta (mBd6,ce5,mPh2).
333 defesaindireta (mBd6,mPe5,cf4).
334 defesaindireta (mBd6,mPe5,cg3).
335 defesaindireta (mBd6,mPe5,ch2).
336 defesaindireta (mBd6,mPe5,mPh2).
337 defesaindireta (mKg1,cf1,cg2).
338 defesaindireta (mKg1,cf1,mPg2).
339 defesaindireta (mKg1,cg2,cf1).
340 defesaindireta (mKg1,cg2,ch2).
341 defesaindireta (mKg1,cg2,mPh2).
342 defesaindireta (mKg1,cg2,mRf1).
343 defesaindireta (mKg1,ch2,cg2).
344 defesaindireta (mKg1,ch2,mPg2).
345 defesaindireta (mKg1,mPg2,cf1).
346 defesaindireta (mKg1,mPg2,ch2).
347 defesaindireta (mKg1,mPg2,mPh2).
348 defesaindireta (mKg1,mPg2,mRf1).
349 defesaindireta (mKg1,mPh2,cg2).
350 defesaindireta (mKg1,mPh2,mPg2).
351 defesaindireta (mKg1,mRf1,cg2).
352 defesaindireta (mKg1,mRf1,mPg2).
353 defesaindireta (mQe2,ca2,cg2).
354 defesaindireta (mQe2,ca2,mPg2).
355 defesaindireta (mQe2,cb5,ca6).
356 defesaindireta (mQe2,cb5,cf1).
357 defesaindireta (mQe2,cb5,mRf1).
358 defesaindireta (mQe2,ce1,ce5).
359 defesaindireta (mQe2,ce1,mPe5).
360 defesaindireta (mQe2,ce5,ce1).
361 defesaindireta (mQe2,ce5,mRe1).
362 defesaindireta (mQe2,cg2,ca2).
363 defesaindireta (mQe2,cg2,cf1).
364 defesaindireta (mQe2,cg2,ch2).
365 defesaindireta (mQe2,cg2,mPa2).
366 defesaindireta (mQe2,cg2,mPh2).
367 defesaindireta (mQe2,cg2,mRf1).
368 defesaindireta (mQe2,mPa2,cg2).
369 defesaindireta (mQe2,mPa2,mPg2).
370 defesaindireta (mQe2,mPe5,ce1).
371 defesaindireta (mQe2,mPe5,mRe1).
372 defesaindireta (mQe2,mPg2,ca2).
373 defesaindireta (mQe2,mPg2,cf1).
374 defesaindireta (mQe2,mPg2,ch2).
375 defesaindireta (mQe2,mPg2,mPa2).

376 defesaindireta (mQe2,mPg2,mPh2).
 377 defesaindireta (mQe2,mPg2,mRf1).
 378 defesaindireta (mQe2,mRe1,ce5).
 379 defesaindireta (mQe2,mRe1,mPe5).
 380 defesaindireta (mQe2,spb5,ca6).
 381 defesaindireta (mQe2,spb5,cf1).
 382 defesaindireta (mQe2,spb5,mRf1).
 383 defesaindireta (mRe1,ce2,ce3).
 384 defesaindireta (mRe1,ce2,ce4).
 385 defesaindireta (mRe1,ce2,ce5).
 386 defesaindireta (mRe1,ce2,mPe5).
 387 defesaindireta (mRe1,cf1,cg1).
 388 defesaindireta (mRe1,cf1,mKg1).
 389 defesaindireta (mRe1,mQe2,ce3).
 390 defesaindireta (mRe1,mQe2,ce4).
 391 defesaindireta (mRe1,mQe2,ce5).
 392 defesaindireta (mRe1,mQe2,mPe5).
 393 defesaindireta (mRe1,mRf1,cg1).
 394 defesaindireta (mRe1,mRf1,mKg1).
 395 defesaindireta (mRf1,ce1,ca1).
 396 defesaindireta (mRf1,ce1,cb1).
 397 defesaindireta (mRf1,ce1,cc1).
 398 defesaindireta (mRf1,ce1,cd1).
 399 defesaindireta (mRf1,ce1,cg1).
 400 defesaindireta (mRf1,ce1,mKg1).
 401 defesaindireta (mRf1,cg1,ce1).
 402 defesaindireta (mRf1,cg1,ch1).
 403 defesaindireta (mRf1,cg1,mRe1).
 404 defesaindireta (mRf1,mKg1,ce1).
 405 defesaindireta (mRf1,mKg1,ch1).
 406 defesaindireta (mRf1,mKg1,mRe1).
 407 defesaindireta (mRf1,mRe1,ca1).
 408 defesaindireta (mRf1,mRe1,cb1).
 409 defesaindireta (mRf1,mRe1,cc1).
 410 defesaindireta (mRf1,mRe1,cd1).
 411 defesaindireta (mRf1,mRe1,cg1).
 412 defesaindireta (mRf1,mRe1,mKg1).
 413 defesaindireta (null , null , null).
 414 defesaindireta (sbd7,cc6,cb5).
 415 defesaindireta (sbd7,cc6,ce8).
 416 defesaindireta (sbd7,cc6,spb5).
 417 defesaindireta (sbd7,cc6,sre8).
 418 defesaindireta (sbd7,cc8,ce6).
 419 defesaindireta (sbd7,cc8,spe6).
 420 defesaindireta (sbd7,ce6,cc8).
 421 defesaindireta (sbd7,ce6,sgc8).
 422 defesaindireta (sbd7,ce8,cc6).
 423 defesaindireta (sbd7,ce8,spc6).
 424 defesaindireta (sbd7,spc6,cb5).
 425 defesaindireta (sbd7,spc6,ce8).
 426 defesaindireta (sbd7,spc6,spb5).
 427 defesaindireta (sbd7,spc6,sre8).
 428 defesaindireta (sbd7,spe6,cc8).
 429 defesaindireta (sbd7,spe6,sgc8).
 430 defesaindireta (sbd7,sgc8,ce6).
 431 defesaindireta (sbd7,sgc8,spe6).
 432 defesaindireta (sbd7,sre8,cc6).
 433 defesaindireta (sbd7,sre8,spc6).
 434 defesaindireta (skf7,cg7,cf6).
 435 defesaindireta (skf7,cg7,cg8).
 436 defesaindireta (skf7,cg7,snf6).
 437 defesaindireta (skf7,cg7,sng8).
 438 defesaindireta (skf7,cg8,cg7).
 439 defesaindireta (skf7,cg8,spg7).
 440 defesaindireta (skf7,sng8,cg7).
 441 defesaindireta (skf7,sng8,spg7).
 442 defesaindireta (skf7,spg7,cf6).
 443 defesaindireta (skf7,spg7,cg8).
 444 defesaindireta (skf7,spg7,snf6).
 445 defesaindireta (skf7,spg7,sng8).
 446 defesaindireta (sqc8,cc6,cc4).
 447 defesaindireta (sqc8,cc6,cc5).
 448 defesaindireta (sqc8,cc6,cd7).
 449 defesaindireta (sqc8,cc6,sbd7).
 450 defesaindireta (sqc8,cd7,ce6).
 451 defesaindireta (sqc8,cd7,spe6).
 452 defesaindireta (sqc8,cd8,ce8).
 453 defesaindireta (sqc8,cd8,sre8).
 454 defesaindireta (sqc8,sbd7,ce6).
 455 defesaindireta (sqc8,sbd7,spe6).
 456 defesaindireta (sqc8,spc6,cc4).
 457 defesaindireta (sqc8,spc6,cc5).
 458 defesaindireta (sqc8,spc6,cd7).
 459 defesaindireta (sqc8,spc6,sbd7).
 460 defesaindireta (sqc8,srd8,ce8).
 461 defesaindireta (sqc8,srd8,sre8).
 462 defesaindireta (srd8,cc8,ca8).
 463 defesaindireta (srd8,cc8,cb8).
 464 defesaindireta (srd8,cc8,ce8).
 465 defesaindireta (srd8,cc8,sre8).
 466 defesaindireta (srd8,ce8,cc8).
 467 defesaindireta (srd8,ce8,cf8).
 468 defesaindireta (srd8,ce8,cg8).
 469 defesaindireta (srd8,ce8,sng8).
 470 defesaindireta (srd8,ce8,sgc8).
 471 defesaindireta (srd8,sgc8,ca8).
 472 defesaindireta (srd8,sgc8,cb8).
 473 defesaindireta (srd8,sgc8,ce8).
 474 defesaindireta (srd8,sgc8,sre8).

475 defesaindireta (srd8,sre8,cc8).
476 defesaindireta (srd8,sre8,cf8).
477 defesaindireta (srd8,sre8,cg8).
478 defesaindireta (srd8,sre8,sng8).
479 defesaindireta (srd8,sre8,sqc8).
480 defesaindireta (sre8,cd8,cc8).
481 defesaindireta (sre8,cd8,cg8).
482 defesaindireta (sre8,cd8,sng8).
483 defesaindireta (sre8,cd8,sqc8).
484 defesaindireta (sre8,cg8,cd8).
485 defesaindireta (sre8,cg8,ch8).
486 defesaindireta (sre8,cg8,srd8).
487 defesaindireta (sre8,sng8,cd8).
488 defesaindireta (sre8,sng8,ch8).
489 defesaindireta (sre8,sng8,srd8).
490 defesaindireta (sre8,srd8,cc8).
491 defesaindireta (sre8,srd8,cg8).
492 defesaindireta (sre8,srd8,sng8).
493 defesaindireta (sre8,srd8,sqc8).
494 defesapossivel (mBb3,ca2,cb1).
495 defesapossivel (mBb3,cc2,cb1).
496 defesapossivel (mBb3,cc2,cd3).
497 defesapossivel (mBb3,cc2,ce4).
498 defesapossivel (mBb3,cc2,cf5).
499 defesapossivel (mBb3,cc2,mPf5).
500 defesapossivel (mBb3,cc4,cd3).
501 defesapossivel (mBb3,cc4,ce2).
502 defesapossivel (mBb3,cc4,mQe2).
503 defesapossivel (mBb3,cd1,ce2).
504 defesapossivel (mBb3,cd1,mQe2).
505 defesapossivel (mBb3,cd5,ce4).
506 defesapossivel (mBb3,cd5,cf3).
507 defesapossivel (mBb3,cd5,cg2).
508 defesapossivel (mBb3,cd5,mPg2).
509 defesapossivel (mBb3,ce6,cf5).
510 defesapossivel (mBb3,ce6,mPf5).
511 defesapossivel (mBb3,spe6,mPf5).
512 defesapossivel (mBd6,ca3,cb2).
513 defesapossivel (mBd6,ca3,cc1).
514 defesapossivel (mBd6,cb4,ca5).
515 defesapossivel (mBd6,cb4,cc3).
516 defesapossivel (mBd6,cb4,mPc3).
517 defesapossivel (mBd6,cb8,ca7).
518 defesapossivel (mBd6,cc5,ca7).
519 defesapossivel (mBd6,cc5,cb6).
520 defesapossivel (mBd6,cc5,cd4).
521 defesapossivel (mBd6,cc5,mPd4).
522 defesapossivel (mBd6,cc7,ca5).
523 defesapossivel (mBd6,cc7,cb6).
524 defesapossivel (mBd6,ce5,cd4).
525 defesapossivel (mBd6,ce5,mPd4).
526 defesapossivel (mBd6,mPe5,mPd4).
527 defesapossivel (mKg1,cf1,ce1).
528 defesapossivel (mKg1,cf1,ce2).
529 defesapossivel (mKg1,cf1,mQe2).
530 defesapossivel (mKg1,cf1,mRe1).
531 defesapossivel (mKg1,cf2,ce1).
532 defesapossivel (mKg1,cf2,ce2).
533 defesapossivel (mKg1,cf2,ce3).
534 defesapossivel (mKg1,cf2,cf3).
535 defesapossivel (mKg1,cf2,cg3).
536 defesapossivel (mKg1,cf2,mQe2).
537 defesapossivel (mKg1,cf2,mRe1).
538 defesapossivel (mKg1,cg2,cf3).
539 defesapossivel (mKg1,cg2,cg3).
540 defesapossivel (mKg1,cg2,ch3).
541 defesapossivel (mKg1,ch2,cg3).
542 defesapossivel (mKg1,ch2,ch3).
543 defesapossivel (mKg1,mRf1,mQe2).
544 defesapossivel (mKg1,mRf1,mRe1).
545 defesapossivel (mPa2,ca3,cb4).
546 defesapossivel (mPc3,cc4,cb5).
547 defesapossivel (mPc3,cc4,cd5).
548 defesapossivel (mPd4,cd5,cc6).
549 defesapossivel (mPd4,cd5,ce6).
550 defesapossivel (mPe5,cf6,ce7).
551 defesapossivel (mPe5,cf6,cg7).
552 defesapossivel (mPf5,ce6,cd7).
553 defesapossivel (mPf5,ce6,cf7).
554 defesapossivel (mPg2,cg3,cf4).
555 defesapossivel (mPg2,cg3,ch4).
556 defesapossivel (mPg2,cg4,cf5).
557 defesapossivel (mPg2,cg4,ch5).
558 defesapossivel (mPh2,ch3,cg4).
559 defesapossivel (mPh2,ch4,cg5).
560 defesapossivel (mQe2,ca2,ca1).
561 defesapossivel (mQe2,ca2,ca3).
562 defesapossivel (mQe2,ca2,cb1).
563 defesapossivel (mQe2,ca2,cb3).
564 defesapossivel (mQe2,ca2,mBb3).
565 defesapossivel (mQe2,cb2,ca1).
566 defesapossivel (mQe2,cb2,ca3).
567 defesapossivel (mQe2,cb2,cb1).
568 defesapossivel (mQe2,cb2,cb3).
569 defesapossivel (mQe2,cb2,cc1).
570 defesapossivel (mQe2,cb2,cc3).
571 defesapossivel (mQe2,cb2,mBb3).
572 defesapossivel (mQe2,cb2,mPc3).
573 defesapossivel (mQe2,cb5,ca5).

574 defesapossivel (mQe2,cb5,cb3).
575 defesapossivel (mQe2,cb5,cb4).
576 defesapossivel (mQe2,cb5,cb6).
577 defesapossivel (mQe2,cb5,cb7).
578 defesapossivel (mQe2,cb5,cb8).
579 defesapossivel (mQe2,cb5,cc5).
580 defesapossivel (mQe2,cb5,cd5).
581 defesapossivel (mQe2,cb5,ce5).
582 defesapossivel (mQe2,cb5,mBb3).
583 defesapossivel (mQe2,cb5,mPe5).
584 defesapossivel (mQe2,cc2,cb1).
585 defesapossivel (mQe2,cc2,cb3).
586 defesapossivel (mQe2,cc2,cc1).
587 defesapossivel (mQe2,cc2,cc3).
588 defesapossivel (mQe2,cc2,cc4).
589 defesapossivel (mQe2,cc2,cf5).
590 defesapossivel (mQe2,cc2,mBb3).
591 defesapossivel (mQe2,cc2,mPc3).
592 defesapossivel (mQe2,cc2,mPf5).
593 defesapossivel (mQe2,cc4,cb3).
594 defesapossivel (mQe2,cc4,cb4).
595 defesapossivel (mQe2,cc4,cc3).
596 defesapossivel (mQe2,cc4,cc5).
597 defesapossivel (mQe2,cc4,cd4).
598 defesapossivel (mQe2,cc4,cd5).
599 defesapossivel (mQe2,cc4,mBb3).
600 defesapossivel (mQe2,cc4,mPc3).
601 defesapossivel (mQe2,cc4,mPd4).
602 defesapossivel (mQe2,cd1,ca1).
603 defesapossivel (mQe2,cd1,cb1).
604 defesapossivel (mQe2,cd1,cb3).
605 defesapossivel (mQe2,cd1,cc1).
606 defesapossivel (mQe2,cd1,cc2).
607 defesapossivel (mQe2,cd1,cd2).
608 defesapossivel (mQe2,cd1,cd3).
609 defesapossivel (mQe2,cd1,cd4).
610 defesapossivel (mQe2,cd1,mBb3).
611 defesapossivel (mQe2,cd1,mPd4).
612 defesapossivel (mQe2,cd2,cc1).
613 defesapossivel (mQe2,cd2,cc3).
614 defesapossivel (mQe2,cd2,cd1).
615 defesapossivel (mQe2,cd2,cd3).
616 defesapossivel (mQe2,cd2,cd4).
617 defesapossivel (mQe2,cd2,ce3).
618 defesapossivel (mQe2,cd2,cf4).
619 defesapossivel (mQe2,cd2,cg5).
620 defesapossivel (mQe2,cd2,mPc3).
621 defesapossivel (mQe2,cd2,mPd4).
622 defesapossivel (mQe2,cd3,cb1).
623 defesapossivel (mQe2,cd3,cc2).
624 defesapossivel (mQe2,cd3,cc3).
625 defesapossivel (mQe2,cd3,cd1).
626 defesapossivel (mQe2,cd3,cd2).
627 defesapossivel (mQe2,cd3,cd4).
628 defesapossivel (mQe2,cd3,ce3).
629 defesapossivel (mQe2,cd3,ce4).
630 defesapossivel (mQe2,cd3,cf3).
631 defesapossivel (mQe2,cd3,cf5).
632 defesapossivel (mQe2,cd3,cg3).
633 defesapossivel (mQe2,cd3,ch3).
634 defesapossivel (mQe2,cd3,mPc3).
635 defesapossivel (mQe2,cd3,mPd4).
636 defesapossivel (mQe2,cd3,mPf5).
637 defesapossivel (mQe2,ce1,ca1).
638 defesapossivel (mQe2,ce1,cb1).
639 defesapossivel (mQe2,ce1,cc1).
640 defesapossivel (mQe2,ce1,cc3).
641 defesapossivel (mQe2,ce1,cd1).
642 defesapossivel (mQe2,ce1,cd2).
643 defesapossivel (mQe2,ce1,cf1).
644 defesapossivel (mQe2,ce1,cf2).
645 defesapossivel (mQe2,ce1,cg3).
646 defesapossivel (mQe2,ce1,ch4).
647 defesapossivel (mQe2,ce1,mPc3).
648 defesapossivel (mQe2,ce1,mRf1).
649 defesapossivel (mQe2,ce3,cc1).
650 defesapossivel (mQe2,ce3,cc3).
651 defesapossivel (mQe2,ce3,cd2).
652 defesapossivel (mQe2,ce3,cd3).
653 defesapossivel (mQe2,ce3,cd4).
654 defesapossivel (mQe2,ce3,cf2).
655 defesapossivel (mQe2,ce3,cf3).
656 defesapossivel (mQe2,ce3,cf4).
657 defesapossivel (mQe2,ce3,cg1).
658 defesapossivel (mQe2,ce3,cg3).
659 defesapossivel (mQe2,ce3,cg5).
660 defesapossivel (mQe2,ce3,ch3).
661 defesapossivel (mQe2,ce3,mKg1).
662 defesapossivel (mQe2,ce3,mPc3).
663 defesapossivel (mQe2,ce3,mPd4).
664 defesapossivel (mQe2,ce4,cb1).
665 defesapossivel (mQe2,ce4,cc2).
666 defesapossivel (mQe2,ce4,cd4).
667 defesapossivel (mQe2,ce4,cd5).
668 defesapossivel (mQe2,ce4,cf4).
669 defesapossivel (mQe2,ce4,cf5).
670 defesapossivel (mQe2,ce4,cg4).
671 defesapossivel (mQe2,ce4,ch4).
672 defesapossivel (mQe2,ce4,mPd4).

673 defesapossivel (mQe2,ce4,mPf5).
674 defesapossivel (mQe2,ce5,cc5).
675 defesapossivel (mQe2,ce5,cd4).
676 defesapossivel (mQe2,ce5,cd5).
677 defesapossivel (mQe2,ce5,cd6).
678 defesapossivel (mQe2,ce5,cf4).
679 defesapossivel (mQe2,ce5,cf5).
680 defesapossivel (mQe2,ce5,cg3).
681 defesapossivel (mQe2,ce5,ch2).
682 defesapossivel (mQe2,ce5,mBd6).
683 defesapossivel (mQe2,ce5,mPd4).
684 defesapossivel (mQe2,ce5,mPf5).
685 defesapossivel (mQe2,ce5,mPh2).
686 defesapossivel (mQe2,cf1,ce1).
687 defesapossivel (mQe2,cf1,cf2).
688 defesapossivel (mQe2,cf1,cf3).
689 defesapossivel (mQe2,cf1,cf4).
690 defesapossivel (mQe2,cf1,cf5).
691 defesapossivel (mQe2,cf1,cg1).
692 defesapossivel (mQe2,cf1,cg2).
693 defesapossivel (mQe2,cf1,mKg1).
694 defesapossivel (mQe2,cf1,mPf5).
695 defesapossivel (mQe2,cf1,mPg2).
696 defesapossivel (mQe2,cf1,mRe1).
697 defesapossivel (mQe2,cf2,cd4).
698 defesapossivel (mQe2,cf2,ce3).
699 defesapossivel (mQe2,cf2,cf3).
700 defesapossivel (mQe2,cf2,cf4).
701 defesapossivel (mQe2,cf2,cf5).
702 defesapossivel (mQe2,cf2,cg1).
703 defesapossivel (mQe2,cf2,cg3).
704 defesapossivel (mQe2,cf2,ch4).
705 defesapossivel (mQe2,cf2,mKg1).
706 defesapossivel (mQe2,cf2,mPd4).
707 defesapossivel (mQe2,cf2,mPf5).
708 defesapossivel (mQe2,cf3,cc3).
709 defesapossivel (mQe2,cf3,cd3).
710 defesapossivel (mQe2,cf3,cd5).
711 defesapossivel (mQe2,cf3,ce3).
712 defesapossivel (mQe2,cf3,ce4).
713 defesapossivel (mQe2,cf3,cf2).
714 defesapossivel (mQe2,cf3,cf4).
715 defesapossivel (mQe2,cf3,cf5).
716 defesapossivel (mQe2,cf3,cg3).
717 defesapossivel (mQe2,cf3,ch3).
718 defesapossivel (mQe2,cf3,mPc3).
719 defesapossivel (mQe2,cf3,mPf5).
720 defesapossivel (mQe2,cg2,cd5).
721 defesapossivel (mQe2,cg2,ce4).
722 defesapossivel (mQe2,cg2,cg1).
723 defesapossivel (mQe2,cg2,cg3).
724 defesapossivel (mQe2,cg2,cg4).
725 defesapossivel (mQe2,cg2,cg5).
726 defesapossivel (mQe2,cg2,cg6).
727 defesapossivel (mQe2,cg2,ch1).
728 defesapossivel (mQe2,cg2,ch3).
729 defesapossivel (mQe2,cg2,mKg1).
730 defesapossivel (mQe2,cg4,cd4).
731 defesapossivel (mQe2,cg4,ce4).
732 defesapossivel (mQe2,cg4,cf4).
733 defesapossivel (mQe2,cg4,cf5).
734 defesapossivel (mQe2,cg4,cg3).
735 defesapossivel (mQe2,cg4,cg5).
736 defesapossivel (mQe2,cg4,cg6).
737 defesapossivel (mQe2,cg4,ch3).
738 defesapossivel (mQe2,cg4,ch4).
739 defesapossivel (mQe2,cg4,mPd4).
740 defesapossivel (mQe2,cg4,mPf5).
741 defesapossivel (mQe2,ch5,cf5).
742 defesapossivel (mQe2,ch5,cg5).
743 defesapossivel (mQe2,ch5,cg6).
744 defesapossivel (mQe2,ch5,ch2).
745 defesapossivel (mQe2,ch5,ch3).
746 defesapossivel (mQe2,ch5,ch4).
747 defesapossivel (mQe2,ch5,mPf5).
748 defesapossivel (mQe2,ch5,mPh2).
749 defesapossivel (mQe2,mPa2,mBb3).
750 defesapossivel (mQe2,mPe5,mBd6).
751 defesapossivel (mQe2,mPe5,mPd4).
752 defesapossivel (mQe2,mPe5,mPf5).
753 defesapossivel (mQe2,mPe5,mPh2).
754 defesapossivel (mQe2,mPg2,mKg1).
755 defesapossivel (mQe2,mRe1,mPc3).
756 defesapossivel (mQe2,mRe1,mRf1).
757 defesapossivel (mQe2,mRf1,mKg1).
758 defesapossivel (mQe2,mRf1,mPf5).
759 defesapossivel (mQe2,mRf1,mPg2).
760 defesapossivel (mQe2,mRf1,mRe1).
761 defesapossivel (mQe2,spb5,mBb3).
762 defesapossivel (mQe2,spb5,mPe5).
763 defesapossivel (mRe1,ca1,ca2).
764 defesapossivel (mRe1,ca1,mPa2).
765 defesapossivel (mRe1,cb1,cb2).
766 defesapossivel (mRe1,cb1,cb3).
767 defesapossivel (mRe1,cb1,mBb3).
768 defesapossivel (mRe1,cc1,cc2).
769 defesapossivel (mRe1,cc1,cc3).
770 defesapossivel (mRe1,cc1,mPc3).
771 defesapossivel (mRe1,cd1,cd2).

772 defesapossivel (mRe1,cd1,cd3).
773 defesapossivel (mRe1,cd1,cd4).
774 defesapossivel (mRe1,cd1,mPd4).
775 defesapossivel (mRe1,ce2,ca2).
776 defesapossivel (mRe1,ce2,cb2).
777 defesapossivel (mRe1,ce2,cc2).
778 defesapossivel (mRe1,ce2,cd2).
779 defesapossivel (mRe1,ce2,cf2).
780 defesapossivel (mRe1,ce2,cg2).
781 defesapossivel (mRe1,ce2,mPa2).
782 defesapossivel (mRe1,ce2,mPg2).
783 defesapossivel (mRe1,cf1,cf2).
784 defesapossivel (mRe1,cf1,cf3).
785 defesapossivel (mRe1,cf1,cf4).
786 defesapossivel (mRe1,cf1,cf5).
787 defesapossivel (mRe1,cf1,mPf5).
788 defesapossivel (mRe1,mQe2,mPa2).
789 defesapossivel (mRe1,mQe2,mPg2).
790 defesapossivel (mRe1,mRf1,mPf5).
791 defesapossivel (mRf1,ce1,ce2).
792 defesapossivel (mRf1,ce1,mQe2).
793 defesapossivel (mRf1,cf2,ce2).
794 defesapossivel (mRf1,cf2,cg2).
795 defesapossivel (mRf1,cf2,mPg2).
796 defesapossivel (mRf1,cf2,mQe2).
797 defesapossivel (mRf1,cf3,cc3).
798 defesapossivel (mRf1,cf3,cd3).
799 defesapossivel (mRf1,cf3,ce3).
800 defesapossivel (mRf1,cf3,cg3).
801 defesapossivel (mRf1,cf3,ch3).
802 defesapossivel (mRf1,cf3,mPc3).
803 defesapossivel (mRf1,cf4,cd4).
804 defesapossivel (mRf1,cf4,ce4).
805 defesapossivel (mRf1,cf4,cg4).
806 defesapossivel (mRf1,cf4,ch4).
807 defesapossivel (mRf1,cf4,mPd4).
808 defesapossivel (mRf1,cf5,ce5).
809 defesapossivel (mRf1,cf5,cg5).
810 defesapossivel (mRf1,cf5,ch5).
811 defesapossivel (mRf1,cf5,mPe5).
812 defesapossivel (mRf1,cg1,cg2).
813 defesapossivel (mRf1,cg1,mPg2).
814 defesapossivel (mRf1,mKg1,mPg2).
815 defesapossivel (mRf1,mPf5,mPe5).
816 defesapossivel (mRf1,mRe1,mQe2).
817 defesapossivel (null, null, null).
818 defesapossivel (sbd7,cc6,ca8).
819 defesapossivel (sbd7,cc6,cb7).
820 defesapossivel (sbd7,cc6,cd5).
821 defesapossivel (sbd7,cc6,ce4).
822 defesapossivel (sbd7,cc6,cf3).
823 defesapossivel (sbd7,cc8,ca6).
824 defesapossivel (sbd7,cc8,cb7).
825 defesapossivel (sbd7,ce6,cc4).
826 defesapossivel (sbd7,ce6,cd5).
827 defesapossivel (sbd7,ce6,cf7).
828 defesapossivel (sbd7,ce6,skf7).
829 defesapossivel (sbd7,ce8,cf7).
830 defesapossivel (sbd7,ce8,skf7).
831 defesapossivel (sbd7,sp6,skf7).
832 defesapossivel (sbd7,sre8,skf7).
833 defesapossivel (skf7,ce8,cd7).
834 defesapossivel (skf7,ce8,cd8).
835 defesapossivel (skf7,ce8,sbd7).
836 defesapossivel (skf7,ce8,srd8).
837 defesapossivel (skf7,cg7,ch6).
838 defesapossivel (skf7,cg7,ch7).
839 defesapossivel (skf7,cg7,ch8).
840 defesapossivel (skf7,cg7,sph6).
841 defesapossivel (skf7,cg8,ch7).
842 defesapossivel (skf7,cg8,ch8).
843 defesapossivel (skf7,spg7,sph6).
844 defesapossivel (skf7,sre8,sbd7).
845 defesapossivel (skf7,sre8,srd8).
846 defesapossivel (snf6,cd5,cb4).
847 defesapossivel (snf6,cd5,cb6).
848 defesapossivel (snf6,cd5,cc7).
849 defesapossivel (snf6,cd5,ce3).
850 defesapossivel (snf6,cd5,ce7).
851 defesapossivel (snf6,cd5,cf4).
852 defesapossivel (snf6,cd7,cb6).
853 defesapossivel (snf6,cd7,cb8).
854 defesapossivel (snf6,cd7,cc5).
855 defesapossivel (snf6,cd7,cf8).
856 defesapossivel (snf6,ce4,cc5).
857 defesapossivel (snf6,ce4,cd2).
858 defesapossivel (snf6,ce4,cf2).
859 defesapossivel (snf6,ce4,cg3).
860 defesapossivel (snf6,ce4,cg5).
861 defesapossivel (snf6,ce8,cc7).
862 defesapossivel (snf6,ce8,cg7).
863 defesapossivel (snf6,ce8,spg7).
864 defesapossivel (snf6,cg4,ce3).
865 defesapossivel (snf6,cg4,cf2).
866 defesapossivel (snf6,cg4,ch6).
867 defesapossivel (snf6,cg4,sph6).
868 defesapossivel (snf6,cg8,ce7).
869 defesapossivel (snf6,cg8,ch6).
870 defesapossivel (snf6,cg8,sph6).

871 defesapossivel (snf6, ch5, cf4).
872 defesapossivel (snf6, ch5, cg3).
873 defesapossivel (snf6, ch5, cg7).
874 defesapossivel (snf6, ch5, spg7).
875 defesapossivel (snf6, ch7, cf8).
876 defesapossivel (snf6, ch7, cg3).
877 defesapossivel (snf6, sng8, sph6).
878 defesapossivel (snf6, sre8, spg7).
879 defesapossivel (sng8, ce7, cc6).
880 defesapossivel (sng8, ce7, cc8).
881 defesapossivel (sng8, ce7, cd5).
882 defesapossivel (sng8, ce7, cg6).
883 defesapossivel (sng8, ce7, spc6).
884 defesapossivel (sng8, ce7, sqc8).
885 defesapossivel (sng8, cf6, cd5).
886 defesapossivel (sng8, cf6, cd7).
887 defesapossivel (sng8, cf6, ce4).
888 defesapossivel (sng8, cf6, ce8).
889 defesapossivel (sng8, cf6, cg4).
890 defesapossivel (sng8, cf6, ch5).
891 defesapossivel (sng8, cf6, ch7).
892 defesapossivel (sng8, cf6, sbd7).
893 defesapossivel (sng8, cf6, sre8).
894 defesapossivel (sng8, ch6, cf7).
895 defesapossivel (sng8, ch6, cg4).
896 defesapossivel (sng8, ch6, skf7).
897 defesapossivel (sng8, snf6, sbd7).
898 defesapossivel (sng8, snf6, sre8).
899 defesapossivel (sng8, sph6, skf7).
900 defesapossivel (spa4, ca3, cb2).
901 defesapossivel (spa4, cb3, ca2).
902 defesapossivel (spa4, cb3, cc2).
903 defesapossivel (spb5, cb4, ca3).
904 defesapossivel (spb5, cb4, cc3).
905 defesapossivel (spc6, cc5, cb4).
906 defesapossivel (spc6, cc5, cd4).
907 defesapossivel (spe6, cf5, ce4).
908 defesapossivel (spe6, cf5, cg4).
909 defesapossivel (spg7, cg5, cf4).
910 defesapossivel (spg7, cg5, ch4).
911 defesapossivel (spg7, cg6, cf5).
912 defesapossivel (spg7, cg6, ch5).
913 defesapossivel (sph6, ch5, cg4).
914 defesapossivel (sqc8, ca6, ca4).
915 defesapossivel (sqc8, ca6, ca5).
916 defesapossivel (sqc8, ca6, ca7).
917 defesapossivel (sqc8, ca6, ca8).
918 defesapossivel (sqc8, ca6, cb5).
919 defesapossivel (sqc8, ca6, cb6).
920 defesapossivel (sqc8, ca6, spa4).
921 defesapossivel (sqc8, ca6, spb5).
922 defesapossivel (sqc8, ca8, ca4).
923 defesapossivel (sqc8, ca8, ca5).
924 defesapossivel (sqc8, ca8, ca6).
925 defesapossivel (sqc8, ca8, ca7).
926 defesapossivel (sqc8, ca8, spa4).
927 defesapossivel (sqc8, cb7, ca7).
928 defesapossivel (sqc8, cb7, ca8).
929 defesapossivel (sqc8, cb7, cb5).
930 defesapossivel (sqc8, cb7, cb6).
931 defesapossivel (sqc8, cb7, cb8).
932 defesapossivel (sqc8, cb7, cc7).
933 defesapossivel (sqc8, cb7, spb5).
934 defesapossivel (sqc8, cb8, ca7).
935 defesapossivel (sqc8, cb8, cb5).
936 defesapossivel (sqc8, cb8, cb6).
937 defesapossivel (sqc8, cb8, cb7).
938 defesapossivel (sqc8, cb8, cc7).
939 defesapossivel (sqc8, cb8, spb5).
940 defesapossivel (sqc8, cc6, ca6).
941 defesapossivel (sqc8, cc6, ca8).
942 defesapossivel (sqc8, cc6, cb5).
943 defesapossivel (sqc8, cc6, cb6).
944 defesapossivel (sqc8, cc6, cd5).
945 defesapossivel (sqc8, cc6, ce4).
946 defesapossivel (sqc8, cc6, cf3).
947 defesapossivel (sqc8, cc6, spb5).
948 defesapossivel (sqc8, cc7, ca5).
949 defesapossivel (sqc8, cc7, ca7).
950 defesapossivel (sqc8, cc7, cb6).
951 defesapossivel (sqc8, cc7, cb7).
952 defesapossivel (sqc8, cc7, cb8).
953 defesapossivel (sqc8, cd7, ca7).
954 defesapossivel (sqc8, cd7, cb7).
955 defesapossivel (sqc8, cd7, cc6).
956 defesapossivel (sqc8, cd7, cc7).
957 defesapossivel (sqc8, cd7, cd8).
958 defesapossivel (sqc8, cd7, ce7).
959 defesapossivel (sqc8, cd7, ce8).
960 defesapossivel (sqc8, cd7, cf7).
961 defesapossivel (sqc8, cd7, skf7).
962 defesapossivel (sqc8, cd7, spc6).
963 defesapossivel (sqc8, cd7, srd8).
964 defesapossivel (sqc8, cd7, sre8).
965 defesapossivel (sqc8, cd8, ca5).
966 defesapossivel (sqc8, cd8, cb6).
967 defesapossivel (sqc8, cd8, cc7).
968 defesapossivel (sqc8, cd8, cd7).
969 defesapossivel (sqc8, cd8, ce7).

```

970 defesapossivel (sqc8,cd8,cf6).
971 defesapossivel (sqc8,cd8,sbd7).
972 defesapossivel (sqc8,cd8,snf6).
973 defesapossivel (sqc8,sbd7,skf7).
974 defesapossivel (sqc8,sbd7,spc6).
975 defesapossivel (sqc8,sbd7,srd8).
976 defesapossivel (sqc8,sbd7,sre8).
977 defesapossivel (sqc8,spc6,spb5).
978 defesapossivel (sqc8,srd8,sbd7).
979 defesapossivel (sqc8,srd8,snf6).
980 defesapossivel (srd8,cc8,cc6).
981 defesapossivel (srd8,cc8,cc7).
982 defesapossivel (srd8,cc8,spc6).
983 defesapossivel (srd8,cd7,ca7).
984 defesapossivel (srd8,cd7,cb7).
985 defesapossivel (srd8,cd7,cc7).
986 defesapossivel (srd8,cd7,ce7).
987 defesapossivel (srd8,cd7,cf7).
988 defesapossivel (srd8,cd7,skf7).
989 defesapossivel (srd8,ce8,ce6).
990 defesapossivel (srd8,ce8,ce7).
991 defesapossivel (srd8,ce8,spe6).
992 defesapossivel (srd8,sbd7,skf7).
993 defesapossivel (srd8,sqc8,spc6).
994 defesapossivel (srd8,sre8,spe6).
995 defesapossivel (sre8,cd8,cd7).
996 defesapossivel (sre8,cd8,sbd7).
997 defesapossivel (sre8,ce6,cf6).
998 defesapossivel (sre8,ce6,snf6).
999 defesapossivel (sre8,ce7,cd7).
1000 defesapossivel (sre8,ce7,cf7).
1001 defesapossivel (sre8,ce7,sbd7).
1002 defesapossivel (sre8,ce7,skf7).
1003 defesapossivel (sre8,cf8,cf7).
1004 defesapossivel (sre8,cf8,skf7).
1005 defesapossivel (sre8,cg8,cg7).
1006 defesapossivel (sre8,cg8,spg7).
1007 defesapossivel (sre8,sng8,spg7).
1008 defesapossivel (sre8,spe6,snf6).
1009 defesapossivel (sre8,srd8,sbd7).
1010 desprotegidapormim(C,F):- (meu(C);casa(C)), minhasdefesas(X,m), not(member(C,X)).
1011 desprotegidaporti(C):- (seu(C);casa(C)), suasdefesas(X), not(member(C,X)).
1012 k(mKg1).
1013 k(null).
1014 k(skf7).
1015 leste(X,Y):-oeste(Y,X).
1016 meu(mBb3).
1017 meu(mBd6).
1018 meu(mKg1).
1019 meu(mPa2).
1020 meu(mPc3).
1021 meu(mPd4).
1022 meu(mPe5).
1023 meu(mPf5).
1024 meu(mPg2).
1025 meu(mPh2).
1026 meu(mQe2).
1027 meu(mRe1).
1028 meu(mRf1).
1029 minhasdefesas(Casas,F) :- findall(C,(meu(Y), (defende(Y,C);defesaindireta(Y,F,C))), not(Y=F)), Casas).
1030 move(mBb3,ca4).
1031 move(mBb3,cc2).
1032 move(mBb3,cc4).
1033 move(mBb3,cd1).
1034 move(mBb3,cd5).
1035 move(mBb3,ce6).
1036 move(mBd6,ca3).
1037 move(mBd6,cb4).
1038 move(mBd6,cb8).
1039 move(mBd6,cc5).
1040 move(mBd6,cc7).
1041 move(mBd6,ce7).
1042 move(mBd6,cf8).
1043 move(mKg1,cf2).
1044 move(mKg1,ch1).
1045 move(mPa2,ca3).
1046 move(mPc3,cc4).
1047 move(mPd4,cd5).
1048 move(mPe5,cf6).
1049 move(mPf5,ce6).
1050 move(mPg2,cg3).
1051 move(mPg2,cg4).
1052 move(mPh2,ch3).
1053 move(mPh2,ch4).
1054 move(mQe2,cb2).
1055 move(mQe2,cb5).
1056 move(mQe2,cc2).
1057 move(mQe2,cc4).
1058 move(mQe2,cd1).
1059 move(mQe2,cd2).
1060 move(mQe2,cd3).
1061 move(mQe2,ce3).
1062 move(mQe2,ce4).
1063 move(mQe2,cf2).
1064 move(mQe2,cf3).
1065 move(mQe2,cg4).
1066 move(mQe2,ch5).
1067 move(mRe1,ca1).
1068 move(mRe1,cb1).

```

```

1069 move(mRe1,cc1).
1070 move(mRe1,cd1).
1071 move(mRf1,cf2).
1072 move(mRf1,cf3).
1073 move(mRf1,cf4).
1074 move(null,null).
1075 move(skf7,ce7).
1076 move(skf7,cf8).
1077 move(skf7,cg6).
1078 move(snf6,cd5).
1079 move(snf6,ce4).
1080 move(snf6,cg4).
1081 move(snf6,ch5).
1082 move(snf6,ch7).
1083 move(sng8,ce7).
1084 move(spa4,ca3).
1085 move(spa4,cb3).
1086 move(spb5,cb4).
1087 move(spc6,cc5).
1088 move(spe6,cf5).
1089 move(spg7,cg5).
1090 move(spg7,cg6).
1091 move(sph6,ch5).
1092 move(sqc8,ca6).
1093 move(sqc8,ca8).
1094 move(sqc8,cb7).
1095 move(sqc8,cb8).
1096 move(sqc8,cc7).
1097 move(sre8,ce7).
1098 move(sre8,cf8).
1099 n(null).
1100 n(snf6).
1101 n(sng8).
1102 ne(X,Y):-so(Y,X).
1103 no(X,Y):-se(Y,X).
1104 norte(X,Y):-sul(Y,X).
1105 oeste(mBb3,mPc3).
1106 oeste(mBd6,spe6).
1107 oeste(mPe5,mPf5).
1108 oeste(mPg2,mPh2).
1109 oeste(mRe1,mRf1).
1110 oeste(mRf1,mKg1).
1111 oeste(null,null).
1112 oeste(skf7,spg7).
1113 oeste(spc6,mBd6).
1114 oeste(spe6,snf6).
1115 oeste(sqc8,srd8).
1116 oeste(srd8,sre8).
1117 p(mPa2).
1118 p(mPc3).
1119 p(mPd4).
1120 p(mPe5).
1121 p(mPf5).
1122 p(mPg2).
1123 p(mPh2).
1124 p(null).
1125 p(spa4).
1126 p(spb5).
1127 p(spc6).
1128 p(spe6).
1129 p(spg7).
1130 p(sph6).
1131 protegidapormim(C,F):- (meu(C);casa(C)), minhasdefesas(X,F), member(C,X).
1132 protegidaporti(C):- (seu(C);casa(C)), suasdefesas(X), member(C,X).
1133 protmim(null).
1134 protti(null).
1135 q(mQe2).
1136 q(null).
1137 q(sqc8).
1138 r(mRe1).
1139 r(mRf1).
1140 r(null).
1141 r(srd8).
1142 r(sre8).
1143 se(mBb3,spa4).
1144 se(mPe5,mBd6).
1145 se(mPf5,spe6).
1146 se(mRf1,mQe2).
1147 se(sbd7,sqc8).
1148 se(skf7,sre8).
1149 se(spe6,sbd7).
1150 se(sph6,spg7).
1151 seu(sbd7).
1152 seu(skf7).
1153 seu(snf6).
1154 seu(sng8).
1155 seu(spa4).
1156 seu(spb5).
1157 seu(spc6).
1158 seu(spe6).
1159 seu(spg7).
1160 seu(sph6).
1161 seu(sqc8).
1162 seu(srd8).
1163 seu(sre8).
1164 so(mKg1,mPh2).
1165 so(mPa2,mBb3).
1166 so(mPc3,mPd4).
1167 so(mPd4,mPe5).

```

```
1168 so(mPe5,snf6).
1169 so(mRf1,mPg2).
1170 so(sbd7,sre8).
1171 so(skf7,sng8).
1172 so(snf6,spg7).
1173 so(spa4,spb5).
1174 so(spb5,spc6).
1175 so(spc6,sbd7).
1176 so(spe6,skf7).
1177 suasdefesas(Casas) :- findall(C,(seu(Y), defende(Y,C)), Casas).
1178 sul(mBd6,sbd7).
1179 sul(mKg1,mPg2).
1180 sul(mPe5,spe6).
1181 sul(mPf5,snf6).
1182 sul(mRe1,mQe2).
1183 sul(null,null).
1184 sul(sbd7,srd8).
1185 sul(snf6,skf7).
1186 sul(spg7,sng8).
1187 xequemate(null,null).
```

Listagem C.1: Base de fatos Prolog

Apêndice D: Outros HLPs

```

1 PATTERN "LINHARES_01_07"
2 AUTHOR "Luis Bueno"
3 DESCRIPTION "Checkmate in one ply"
4 WEIGHT 1.0
5 SCENARIO
6   F1 POSSIBLEATTACK (S1,k1)
7   F2 DEFENDS S1
8 PRECONDITION
9   DIRECTDEFENSES(S1,OPONENT) = 0
10 TACTICALS
11   F1,S1
12 POSTCONDITION
13   MOVESOF(k1) = 0;
14
15 PATTERN "Linhares_02_04"
16 AUTHOR "Luis Bueno"
17 DESCRIPTION "Checkmate with discovered attack"
18 WEIGHT 1.0
19 SCENARIO
20   F1 INDIRECTATTACK (F2,k1)
21   F2 MOVE S1
22 PRECONDITION
23   MOVESOF(k1) <= 1
24 TACTICALS
25   F2,S1
26 POSTCONDITION
27   MOVESOF(k1) = 0;
28
29 PATTERN "LINHARES_03_16"
30 AUTHOR "Luis Bueno"
31 DESCRIPTION "Blocked pawns"
32 WEIGHT 1.0
33 ACTORS K1,k1
34 SCENARIO
35   p1 DEFENDS p2
36   p2 DEFENDS p3
37   p3 DEFENDS p4
38   {p1 DEFENDS p5
39   p5 DEFENDS p4}
40 PRECONDITION
41   MOVESOF(P) = 0 AND
42   (KINGPATHTO(K1,p1) = 0 OR
43   INTHE SQUARE(k1,p1));
44
45 PATTERN "LINHARES_03_16_ALT"
46 AUTHOR "Luis Bueno"
47 DESCRIPTION "Blocked pawns"
48 WEIGHT -999.0
49 ACTORS K1,k1
50 SCENARIO
51   >p1< DEFENDE S1
52 PRECONDITION
53   KINGPATHTO(K1,p1) = 0 AND
54   INTHE SQUARE(k1,p1) AND
55   KINGOPPOSITION(K1,k1) = FALSE AND
56   EXISTPATTERN("BLOQUEIO",OPONENT);
57
58 PATTERN "LINHARES_05_09"
59 AUTHOR "Luis Bueno"
60 DESCRIPTION "Absolute fork"
61 WEIGHT 1.0
62 SCENARIO
63   F1 POSSIBLEATTACK (>s1<,k1)
64   F1 POSSIBLEATTACK (>s1<,>f1<);
65
66 PATTERN "LINHARES_11_19"
67 AUTHOR "Luis Bueno"
68 DESCRIPTION "Distant passed pawn"
69 WEIGHT 1.0
70 ACTORS K1,k1
71 SETOFPIECES KPkP
72 SCENARIO
73   P1 DEFENDS S1
74   P2 DEFENDS S2
75 PRECONDITION
76   MAXDISTANCE(K1,P2) = MAXDISTANCE(k1,P2) AND
77   SENTRY(P2) = FALSE AND
78   RAM(P2) = FALSE AND
79   BLOCKEDPAWNS(OPONENT) AND
80   (RAM(P1) OR SENTRY(P1)) AND
81   KINGOPPOSITION(K1,k1);
82
83 PATTERN "LINHARES_11_19a"
84 AUTHOR "Luis Bueno"
85 DESCRIPTION "Distant passed pawn"

```

```

86 WEIGHT 1.0
87 ACTORS K1,k1
88 SETOFPIECES KPkp
89 SCENARIO
90   P1 DEFENDS S1
91   P2 DEFENDS S2
92 PRECONDITION
93   INTESQUARE(k1,P2) AND
94   SENTRY(P2)= 0 AND
95   RAM(P2) = 0 AND
96   BLOCKEDPAWNS(OPPONENT) AND
97   (RAM(P1) > 0 OR SENTRY(P1) > 0) AND
98   COLDISTANCE(P1,P2) > 3;
99
100 PATTERN "LINHARES_12_13"
101 AUTHOR "Luis Bueno"
102 DESCRIPTION "Unmovable pawns"
103 WEIGHT 1.0
104 ACTORS k1,K1
105 SETOFPIECES KPkp
106 SCENARIO
107   K1 MOVE S1
108 PRECONDITION
109   BLOCKEDPAWNS(ME) AND
110   BLOCKEDPAWNS(OPPONENT) AND
111   KINGOPPOSITION(K1,k1);
112
113 PATTERN "Linhares_14_17"
114 AUTHOR "Luis Bueno"
115 DESCRIPTION "Blocked pawns and weak bishops"
116 WEIGHT 1.0
117 ACTORS K1, k1
118 SETOFPIECES KBPkbp
119 SCENARIO
120   B1 POSSIBLEDEFENSE (S3,P1)
121   b1 POSSIBLEDEFENSE (S4,p1)
122 PRECONDITION
123   BLOCKEDPAWNS(ME) AND
124   BLOCKEDPAWNS(OPPONENT) AND
125   SQUARECOLOROF(B1) != SQUARECOLOROF(f) AND
126   SQUARECOLOROF(b1) != SQUARECOLOROF(F);
127
128 PATTERN "LINHARES_15_18"
129 AUTHOR "Luis Bueno"
130 DESCRIPTION "Unmovable pawns and active bishop"
131 WEIGHT 1.0
132 ACTORS b1
133 SETOFPIECES KBPkbp
134 SCENARIO
135   B1 POSSIBLEATTACK (S1,k1)
136 PRECONDITION
137   BLOCKEDPAWNS(ME) AND
138   BLOCKEDPAWNS(OPPONENT) AND
139   SQUARECOLOROF(b1) != SQUARECOLOROF(P) E
140   SQUARECOLOROF(b1) != SQUARECOLOROF(K);

```

Listagem D.1: HLPs representados