



UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Engenharia Elétrica e de Computação

Suelen Mapa de Paula

Um Mecanismo de Simulação Mental para a Fundamentação de Sentenças Simbólicas

Campinas

2019

Suelen Mapa de Paula

Um Mecanismo de Simulação Mental para a Fundamentação de Sentenças Simbólicas

Tese apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Doutor em Engenharia Elétrica, na Área de Engenharia de Computação.

Orientador: Prof. Dr. Ricardo Ribeiro Gudwin

Este trabalho corresponde à versão final da tese defendida pela aluna Suelen Mapa de Paula, e orientada pelo Prof. Dr. Ricardo Ribeiro Gudwin.

Campinas

2019

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Luciana Pietrosanto Milla - CRB 8/8129

P281m Paula, Suelen Mapa de, 1982-
Um mecanismo de simulação mental para a fundamentação de sentenças simbólicas / Suelen Mapa de Paula. – Campinas, SP : [s.n.], 2019.

Orientador: Ricardo Ribeiro Gudwin.
Tese (doutorado) – Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Sistemas inteligentes. 2. Sinais e símbolos. 3. Emergências. 4. Inteligência artificial. 5. Comunicação. I. Gudwin, Ricardo Ribeiro, 1967-. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: A Mental simulation mechanism for the grounding of symbolic sentences

Palavras-chave em inglês:

Intelligent systems

Signs e symbols

Emergence

Artificial intelligence

Communication

Área de concentração: Engenharia de Computação

Titulação: Doutora em Engenharia Elétrica

Banca examinadora:

Ricardo Ribeiro Gudwin [Orientador]

Esther Luna Colombini

Paula Dornhofer Paro Costa

Ângelo Conrado Loula

Ely Edison da Silva Matos

Data de defesa: 16-12-2019

Programa de Pós-Graduação: Engenharia Elétrica

Identificação e informações acadêmicas do(a) aluno(a)

- ORCID do autor: 0000-0002-7872-4020

- Currículo Lattes do autor: <http://lattes.cnpq.br/1858596002181893>

COMISSÃO JULGADORA - TESE DE DOUTORADO

Candidato(a): Suelen Mapa de Paula RA: 123229

Data de defesa: 16 de Dezembro de 2019

Titulo da Tese: "Um Mecanismo de Simulação Mental para a Fundamentação de Sentenças Simbólicas"

Prof. Dr. Ricardo Ribeiro Gudwin (Presidente)

Profa. Dra. Esther Luna Colombini (IC/UNICAMP)

Profa. Dra. Paula Dornhofer Paro Costa (FEEC/UNICAMP)

Profa. Dr. Ângelo Conrado Loula (Universidade Estadual de Feira de Santana)

Profa. Dr. Ely Edison da Silva Matos (Universidade Federal de Juiz de Fora)

A Ata de Defesa, com as respectivas assinaturas dos membros da Comissão Julgadora, encontra-se no SIGA (Sistema de Fluxo de Dissertação/Tese) e na Secretaria de Pós-Graduação da Faculdade de Engenharia Elétrica e de Computação.

*Dedico este trabalho ao Davi, meu filho, por sua chegada motivadora.
E ao Rafael, meu esposo, por seu amor e cuidado.*

Agradecimentos

Para conclusão deste projeto de pesquisa uma longa caminhada foi trilhada. Ela só foi possível porque tive ajuda de muitas pessoas, a quem não poderia deixar de agradecer nesse momento:

A Deus, meu amigo de todas as horas, por me permitir alcançar muito mais do que eu julguei ser capaz.

Ao Rafael que desde sempre me ajudou e incentivou. E ao Davi por ressignificar essa caminhada, tornando-a mais leve. Obrigada por tanto amor. Também amo vocês!

Ao meu orientador, Prof. Dr. Ricardo Ribeiro Gudwin, pela orientação, críticas e paciência. Agradeço também pela oportunidade de estudar em um centro de ensino de excelência.

A minha família pelas orações e apoio incondicional. Agradeço de maneira especial a minha mãe que, com sua serenidade e fé, sempre traz um novo vigor para meus momentos de desânimo. A senhora é e sempre será o meu maior exemplo!

Ao meu amigo Faister Cabrera Carvalho por compartilhar comigo seus conhecimentos e sem os quais não finalizaria este trabalho.

Aos meus amigos do LCA que, de uma forma ou outra, contribuíram com sua amizade e apoio. De maneira especial, agradeço ao Wallace Souza por seu apoio de sempre e pelos momentos de descontração compartilhados. E ao Raphael Voltoline pela ajuda na condução dos experimentos.

Aos amigos Tania Basso, Leandro Piardi, Ana Letícia Lelis, Edér Valdir, Alice Cordovil e Andrei Braga por serem nossa família em Campinas. Sempre levarei vocês no meu coração!

Às profas. Dras. Letícia Rittner e Wu Shin-Ting por todo carinho e cuidado. Agradeço também por cederem seus recursos computacionais para a realização dos experimentos deste trabalho.

À Universidade Estadual de Campinas (UNICAMP).

Ao Instituto Federal de Minas Gerais (IFMG) e colegas de trabalho pelo apoio.

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

*“Nada te perturbe, nada te amedronte. Tudo passa, a paciência tudo alcança. A quem
tem Deus nada falta. Só Deus basta!”
(Santa Teresa de D’Ávila)*

Resumo

Contribuindo com o desenvolvimento da área de pesquisas em linguagem gramatical artificial, o presente trabalho apresenta uma proposta para um mecanismo de simulação mental capaz de atribuir significado a símbolos utilizados em sentenças compostas por diferentes classes de palavras (não somente palavras isoladas). O pressuposto que norteou nossa pesquisa foi: para criar estruturas simbólicas gramaticais, deve existir um tipo de representação intermediária que, durante o processo de categorização, possa mapear a realidade em categorias ontológicas. O mecanismo de simulação mental associa dinamicamente sinais de sensores e atuadores, bem como eventos ocorridos no ambiente a símbolos de diferentes categorias ontológicas. Dessa forma, diversas classes de palavras, tais como substantivos, adjetivos, verbos e advérbios, podem ter seu significado atribuído dinamicamente, propiciando aos sistemas inteligentes a capacidade de aprender o significado de novas palavras, gerar e interpretar sentenças, por meio da interação com outros sistemas e com seu ambiente. Para validar seu funcionamento, diferentes tipos de jogos de linguagem (jogos de adivinhação, de ação e composicionais) foram implementados. Os resultados apurados mostraram que uma população de agentes foi capaz de alcançar sucesso comunicativo e de alinhamento, mesmo lidando com diferentes cenários simulados. Neste caso, os agentes conseguiram interpretar nomes e sentenças empregados para descrever objetos e eventos, manifestando preferências pelos mesmos. Os resultados também demonstraram que os agentes conseguiram alcançar sucesso lexical e sintático, i.e., chegaram a um consenso sobre as funções gramaticais dos símbolos utilizados para formar as sentenças, bem como as regras sintáticas utilizadas para combiná-los. Sendo assim, pode-se concluir que o mecanismo de simulação mental permitiu que os agentes representassem a realidade usando categorias ontológicas e as empregassem para criar e interpretar símbolos isolados e composicionais entre eles, sem depender de recursos adicionais para definir seus significados a priori.

Palavras-chaves: símbolos composicionais; sentenças simbólicas; problema da fundamentação do símbolo; percepção; linguagem; linguagem gramatical artificial; jogos de linguagem.

Abstract

Contributing to the development of artificial grammatical language topic of research, this thesis presents a proposal for a mental simulation mechanism. This mechanism is able to assign meaning to symbols used in sentences composed of different classes of words (not just isolated words). Our main assumption is: to create symbolic grammar structures, during the categorization process, we must have an intermediate representation, which segments reality onto ontological categories. The mental simulation mechanism dynamically associates signals from sensors and actuators, as well as events occurring in the environment to symbols from different ontological categories. In this sense, different classes of words, such as nouns, adjectives, verbs and adverbs, might have their meanings assigned dynamically. As a consequence, this dynamic assignment mechanism provides for the intelligent system the capacity to learn the meaning of new words, generate and interpret sentences, through the interaction with other systems and the environment. To validate the simulation, different language games (guessing, action and compositional games) were implemented. Our findings show that a population of agents was able to achieve communicative and alignment success, even dealing with different simulated scenarios. The agents were able to interpret names and sentences used to describe objects and events, as well as having a preference for them. The results also demonstrated that the agents were able to achieve lexical and syntactic success. That is, they reached a consensus on the grammatical functions of the symbols used to form the sentences, as well as the syntactic rules used to combine them. Thus, we might conclude that the mental simulation mechanism allowed agents to represent reality using ontological categories and used them to create and interpret isolated and compositional symbols between them, without depending on additional resources to define their meanings a priori.

Keywords: compositional symbols; symbolic sentence; symbol grounding problem; perception; language; artificial grammar language; language games.

Lista de ilustrações

1	Hipótese proposta para a fundamentação de símbolos gramaticais . . .	25
2	Relação de associação entre representações simbólicas	32
3	Diagrama de Voronoi de um espaço conceitual de duas dimensões baseado em 5 protótipos.	42
4	Configuração do Experimento Talking Head.	49
5	Protocolo de conversação entre os agentes do experimento <i>Talking Heads</i>	50
6	Cena composta por 4 objetos. Adaptado de Borensztajn e Trehan (2006).	52
7	Diagrama de classes simplificado do <i>framework</i> de simulação mental proposto.	83
8	Processos que compõem o ciclo do mecanismo de simulação mental proposto.	88
9	Detalhamento do processo de descobrir categorias ontológicas.	90
10	Detalhamento do processo de perceber eventos.	91
11	Detalhamento do processo de nomear eventos.	91
12	Detalhamento do processo de interpretar símbolos composicionais.	92
13	O sensoriameto de vários objetos ao longo do tempo e suas transformações espaço-temporal.	93
14	Esquema do Algoritmo de Rastreamento.	95
15	Esquema do Algoritmo de Rastreamento - Continuação do esquema apresentado na Figura 14.	98
16	Análise gramatical e sintática de uma sentença que pode ser gerada pelo pacote de linguagem proposto.	100
17	As estruturas de dados mantidas pelo mecanismo de linguagem: a Árvore de Símbolos e a Tabela de Categorias. Ambas preenchidas com os símbolos da sentença “xuzuli motochi raph xuma talazem”.	103
18	Exemplo 1 - Ajuste do percentual de associação entre entre símbolo + categorias na árvore de símbolos e entre categoria + símbolos na tabela de categorias.	107
19	Exemplo 2 - Ajuste do percentual de associação entre entre símbolo + categorias na árvore de símbolos e entre categoria + símbolos na tabela de categorias.	108
20	Exemplo 3 - Ajuste do percentual de associação entre entre símbolo + categorias na árvore de símbolos e entre categoria + símbolos na tabela de categorias.	109
21	Estrutura das sentenças nominal e verbal previstas no pacote de linguagem.	110

22	Emergência das regras sintáticas de uma sentença verbal.	113
23	Rede ESOM em três momentos diferentes. Em (a) os três primeiros nós são criados (nós 1, 2 e 3); Em (b) três novos nós são adicionados (os nós 4, 5 e 6); Em (c) a ligação mais fraca é removida - Adaptado de Deng e Kasabov (2000).	124
24	Representação gráfica de nosso conjunto de dados sintéticos, composto por 12 classes (com 2 dimensões cada) e 1000 pontos de dados. Cada ponto foi gerado aleatoriamente e um ruído foi aplicado a eles, seguindo uma distribuição normal ($\mu = 0$, $\sigma = 2$).	124
25	Representação gráfica de nosso conjunto de dados sintético apresentado na Figura 24, com cada classe original representada por uma cor distinta.	125
26	Pontos rotulados em tempo real através de uma técnica não-supervisionada (ESOM) e dispostos em um diagrama de Voronoi após o final da 1ª fase de classificação. Cada cor representa um rótulo diferente e os pontos pretos grandes representam os protótipos (nós da rede), determinados pelo ESOM.	125
27	Pontos rotulados em tempo real pelo ESOM e dispostos em um diagrama de Voronoi após o final da 10ª fase de classificação.	126
28	Protocolo do Jogo de Adivinhação estabelecido. Os quadrados representam os dados produzidos pelos agentes. Os textos refere-se aos processos inerentes a cada fase. Os processos podem falhar. Entretanto, existem estratégias de diagnóstico e reparo associadas a cada um deles. Um jogo de adivinhação consiste em milhares dessas interações.	128
29	Sucesso de alinhamento para as estratégias de inibição lateral e jogo de nomeação mínimo. A Figura(a) mostra o sucesso de alinhamento alcançado por nosso método e a Figura (b) mostra o sucesso de alinhamento alcançado por Wellens (2012). Ambas as estratégias de inibição lateral usaram como parâmetro: $\delta_{inc} = \delta_{dec} = 0.1$, $\delta_{inh} = 0.5$ and $s_{initial} = 0.5$	129
30	Lista de Categorias Ontológicas empregadas para a categorização dos componentes da simulação.	133
31	O protocolo do jogo de linguagem implementado. Os quadrados representam os dados produzidos pelos agentes. Os textos referem-se aos processos inerentes a cada fase. Os processos podem falhar. Entretanto, existem estratégias de diagnóstico e reparo associadas a cada um deles. Um jogo de linguagem consiste em milhares dessas interações/duelos.	134
32	Interface da Aplicação do Jogo de Linguagem.	137
33	Aba de inspeção de duelos - Início de um novo duelo linguístico entre Adriano (falante) e Tiago (ouvinte)	138

34	Os diferentes níveis de representação mental gerado pelo mecanismo de simulação mental no jogo de linguagem.	139
35	Sentença Expressada pelo Falante.	140
36	O rastreamento do objeto conceitual círculo 2 na mente do ouvinte.	141
37	A construção do evento na mente do ouvinte.	142
38	A linguagem de um dos agentes após o processamento de 2000 <i>frames</i>	143
39	A linguagem de um dos agentes após o processamento de 10000 <i>frames</i>	144
40	A linguagem de um dos agentes após o processamento de 50000 <i>frames</i>	145
41	Cenário T1 com 2 Agentes: Resultado experimental para estratégia de aprendizado gramatical descrita na seção 6.1.2. Parâmetros utilizados na estratégia: p_a : 1.0, p_{mr} :0.20, p_{mp} : 0.03, p_{rem} : 0.0005, p_{ne} : 0.8. População de 2 agentes e ringue com 5 figuras geométricas distintas. Foram executados aproximadamente 1100 duelos. Em (a) os agentes alcançaram completo sucesso comunicativo e de alinhamento. Em (b) o alinhamento dos nomes empregados para as categorias de objetos também foi completo.	148
42	Cenário T1 com 6 Agentes: Resultado experimental para estratégia de aprendizado gramatical descrita na seção 6.1.2. Parâmetros utilizados na estratégia: p_a : 1.0, p_{mr} :0.20, p_{mp} : 0.03, p_{rem} : 0.0005, p_{ne} : 0.8. População de 6 agentes e ringue com 5 figuras geométricas distintas. Foram executados aproximadamente 8500 duelos. Em (a) os agentes alcançaram completo sucesso comunicativo e de alinhamento. Em (b) o alinhamento dos nomes empregados para as categorias de objetos também foi completo.	149
43	Cenário T2 com 2 Agentes: Resultado experimental para estratégia de aprendizado gramatical (descrita na seção 6.1.2) e para estratégia de aprendizado sintático (descrita na seção 6.2.1). Parâmetros utilizados na estratégia de aprendizado gramatical: p_a : 1.0, p_{mr} :0.20, p_{mp} : 0.03, p_{rem} : 0.0005, p_{ne} : 0.8. População de 2 agentes e ringue com 8 figuras geométricas (4 círculos e 4 triângulos). Foram executados aproximadamente 4000 duelos. Em (a) os agentes alcançaram completo sucesso comunicativo e de alinhamento. Em (b) e (c) o alinhamento dos nomes empregados para as categorias de objetos e propriedades também foi completo. Em (d) também houve completo alinhamento sintático.	151

44	Cenário T2 com 6 Agentes: Resultado experimental para estratégia de aprendizado gramatical (descrita na seção 6.1.2) e para estratégia de aprendizado sintático (descrita na seção 6.2.1). Parâmetros utilizados na estratégia de aprendizado gramatical: p_a : 1.0, p_{mr} :0.20, p_{mp} : 0.03, p_{rem} : 0.0005, p_{ne} : 0.8. População de 6 agentes e ringue com 8 figuras geométricas (4 círculos e 4 triângulos). Foram executados aproximadamente 55000 duelos. Em (a) os agentes alcançaram pleno sucesso comunicativo e de alinhamento. Em (b) e (c) o alinhamento dos nomes empregados para as categorias de objetos e propriedades foi alto, porém o alinhamento dos nomes das propriedades não atingiu 100%. Em (d) houve pleno alinhamento sintático.	153
45	Cenário T2 com 6 Agentes (completo alinhamento de nomes): Resultado experimental para estratégia de aprendizado gramatical (descrita na seção 6.1.2) e para estratégia de aprendizado sintático (descrita na seção 6.2.1). Parâmetros utilizados na estratégia de aprendizado gramatical: p_a : 1.0, p_{mr} :0.20, p_{mp} : 0.03, p_{rem} : 0.0005, p_{ne} : 0.8. População de 6 agentes e ringue com 8 figuras geométricas (4 círculos e 4 triângulos). Foram executados aproximadamente 56500 duelos. Em (a) os agentes alcançaram completo sucesso comunicativo e de alinhamento. Em (b) e (c) o alinhamento dos nomes empregados para as categorias de objetos e propriedades foi completo. Em (d) também houve completo alinhamento sintático.	155
46	Cenário T3 com 2 Agentes: Resultado experimental para estratégia de aprendizado gramatical (descrita na seção 6.1.2) e para estratégia de aprendizado sintático (descrita na seção 6.2.1). Parâmetros utilizados na estratégia de aprendizado gramatical: p_a : 1.0, p_{mr} :0.20, p_{mp} : 0.03, p_{rem} : 0.0005, p_{ne} : 0.8. População de 2 agentes e ringue com 6 figuras geométricas (2 retângulos, 2 quadrados e 2 trapézios). Foram executados aproximadamente 12800 duelos. Em (a) os agentes alcançaram completo sucesso comunicativo e de alinhamento. Em (b) e (c) o alinhamento dos nomes empregados para as categorias de objetos, objetos conceituais, <i>affordances</i> , propriedades e modificadores foi completo. Em (d) também houve completo alinhamento sintático.	157

47	Cenário T3 com 6 Agentes: Resultado experimental para estratégia de aprendizado gramatical (descrita na seção 6.1.2) e para estratégia de aprendizado sintático (descrita na seção 6.2.1). Parâmetros utilizados na estratégia de aprendizado gramatical: p_a : 1.0, p_{mr} :0.20, p_{mp} : 0.03, p_{rem} : 0.0005, p_{ne} : 0.8. População de 6 agentes e ringue com 6 figuras geométricas (2 retângulos, 2 quadrados e 2 trapézios). Foram executados aproximadamente 80000 duelos. Em (a) os agentes alcançaram completo sucesso comunicativo e de alinhamento. Em (b) e (c) o alinhamento dos nomes empregados para as categorias de objetos e propriedades foi completo. Em (d) também houve completo alinhamento sintático.	159
48	Reparticionamento do espaço conceitual.	197
49	Reposicionamento do ponto prototípico no espaço conceitual.	198
50	Classificação dos conceitos de objetos modelados.	200
51	Interface da Aplicação do Jogo de Linguagem.	221
52	Área de Configuração.	222
53	Área de Inspeção da Mentes.	225
54	Apresentação dos objetos categorizados (A), dos eventos percebidos (B) e da linguagem que está em evolução durante o jogo (C).	225
55	Estruturas de dados expandidas. O item da esquerda exibe a tabela de categorias e o item da direita a árvore de símbolos.	226
56	Área de Inspeção. Em (A) aba de duelos e suas informações. Em (B) as informações do duelo corrente. E em (C) as informações do histórico de duelos.	227
57	Área de Inspeção: A aba de objetos brutos.	228
58	A área de console com as sentenças expressas.	228

Lista de tabelas

1	Divisão dos trabalhos pesquisados por temas abordados.	66
2	Classificação dos <i>frameworks</i> de Aplicação Geral.	68
3	Classificação dos <i>frameworks</i> Aplicados à Linguística.	71
4	Trabalhos correlatos apresentados no Capítulo 4.	179

Sumário

1	INTRODUÇÃO	20
1.1	Hipótese de Pesquisa	23
1.2	Objetivos	26
1.3	Metodologia de Trabalho	27
1.4	Lista de Publicações	27
1.5	Organização do Trabalho	28
2	FUNDAMENTAÇÃO TEÓRICA	29
2.1	Ontologias	30
2.2	O Problema da Fundamentação do Símbolo	31
2.3	A Teoria Semiótica de Peirce	33
2.4	A Teoria do Sistema de Símbolos Perceptuais de Barsalou	36
2.5	A Teoria do Espaço Conceitual	38
2.5.1	Categorizações de Propriedades e de Objetos	40
2.5.2	A Teoria Semântica das Classes de Palavras	42
2.5.2.1	Substantivos	43
2.5.2.2	Adjetivos	43
2.5.2.3	Verbos	44
2.5.2.4	Advérbios	45
2.5.2.5	Preposições	45
2.6	Considerações Finais	46
3	JOGOS DE LINGUAGEM	48
3.1	Definição	48
3.2	Jogo de Discriminação	51
3.3	Jogo de Nomeação	52
3.3.1	Sobre o Dicionário Léxico Formado e as Estratégias de Aprendizado	54
3.3.2	Medindo o Sucesso do Sistema de Comunicação que está se Formando	55
3.4	Jogo de Ação	56
3.5	Considerações sobre Jogos de Linguagem Composicionais	59
3.5.1	Símbolos Composicionais que Descrevem Objetos através de seus Domínios Semânticos	60
3.5.1.1	Medindo o Sucesso do Sistema de Comunicação que está se Formando	61
3.5.2	Símbolos Composicionais Formados por Múltiplas Palavras e que Dependem de Regras Gramaticais	62
3.5.2.1	Medindo o Sucesso do Sistema de Comunicação que está se Formando	63

3.6	Considerações Finais.	64
4	TRABALHOS CORRELATOS	66
4.1	Análise Crítica dos Trabalhos Encontrados na Literatura	67
4.2	Considerações Finais.	77
5	MECANISMO COMPUTACIONAL PARA SIMULAÇÃO MENTAL	81
5.1	A Modelagem Computacional para o Mecanismo de Simulação Mental Proposto	82
5.1.1	Sobre os Pacotes de <i>Software</i>	83
5.1.1.1	Pacote <i>Raw</i>	83
5.1.1.2	Pacote <i>Conceptual</i>	84
5.1.1.3	Pacote <i>Ontology</i>	84
5.1.1.4	Pacote <i>Perception</i> .	86
5.1.1.5	Pacote <i>Language</i>	86
5.1.2	Interações dos Pacotes de Software Durante o Ciclo Cognitivo da Simulação Mental	87
5.1.2.1	Sobre a Modelagem de Informações Sensoreadas	89
5.1.2.2	Sobre o Descobrimento de Categorias Ontológicas	89
5.1.2.3	Sobre a Percepção de Eventos	90
5.1.2.4	Sobre a Nomeação de Eventos	91
5.1.2.5	Sobre a Interpretação de Símbolos Composicionais	92
5.2	Simulação Mental: Modo Percepção	93
5.2.1	Algoritmo de Rastreamento	94
5.3	Considerações Finais.	99
6	MECANISMO DE SIMULAÇÃO MENTAL: MODO LINGUAGEM	100
6.1	Como Símbolos Isolados são Aprendidos e Associados às suas Funções Gramaticais?	101
6.1.1	Sobre as Estruturas de Dados Mantidas pelo Mecanismo de Linguagem.	102
6.1.2	Sobre o Aprendizado Gramatical	104
6.2	Como as Regras Sintáticas são Aprendidas para Definir a Estrutura das Sentenças?	108
6.2.1	Sobre o Aprendizado Sintático.	111
6.3	A Geração e Interpretação Simbólica.	115
6.3.1	Algoritmo para Obtenção de Símbolos	116
6.3.2	Algoritmo para Interpretação de Símbolos.	116
6.4	Sobre as Transformações	118
6.4.1	A Transformação de Tempo	118
6.5	Considerações Finais.	119

7	EXPERIMENTOS COMPUTACIONAIS	120
7.1	Experimento 1: A Segmentação dos Espaços Conceituais	120
7.1.1	Espaços Conceituais e Mapas Auto-Organizáveis Evolutivos	121
7.1.2	A Implementação e Funcionamento da Rede Neural ESOM	123
7.1.3	Validação: Particionando Espaços Conceituais com ESOM.	123
7.1.4	Jogo de Linguagem Implementado	126
7.1.5	Resultados e Discussões.	128
7.2	Experimento 2: Validação do Mecanismo de Simulação Mental Atuando nos Modos Percepção e Linguagem.	130
7.2.1	Categorias Ontológicas Modeladas para o Jogo	132
7.2.2	Protocolo do Jogo de Linguagem Implementado	133
7.2.3	Interface Gráfica	137
7.2.3.1	Exemplo de Jogo de Linguagem para Ilustrar Recursos da Interface Gráfica	138
7.2.3.2	A Evolução da Linguagem durante o Jogo de Linguagem	142
7.2.4	Experimento 2: Resultados e Discussões.	145
7.2.4.1	A Configuração Adotada	146
7.2.4.2	Cenário T1	147
7.2.4.3	Cenário T2	150
7.2.4.4	Cenário T3	156
7.3	Considerações Finais.	160
8	CONCLUSÃO	163
8.1	Principais Contribuições deste Trabalho	164
8.2	Perspectivas de Trabalhos Futuros	165
8.3	Considerações Finais.	167
	REFERÊNCIAS	169
	ANEXOS	177
A	SUMÁRIO DE TRABALHOS CORRELATOS	178
B	ASPECTOS DE IMPLEMENTAÇÃO DOS PACOTES DE SOFTWARE	194
B.1	Aspectos de Implementação do Pacote <i>Raw</i>	194
B.2	Aspectos de Implementação do Pacote <i>Conceptual</i>	196
B.3	Aspectos de Implementação do Pacote <i>Ontology</i>	199
B.4	Aspectos de Implementação do Pacote <i>Perception</i>	203
B.5	Aspectos de Implementação do Pacote <i>Language</i>	204

C	DETALHES DOS EXPERIMENTOS REALIZADOS	206
C.1	Configuração das categorias ontológicas empregadas nos experimentos T1, T2 e T3	206
C.2	Interface Gráfica da Simulação	221
C.2.1	A Área de Visualização do Mundo	221
C.2.2	A Área de Configuração.	222
C.2.3	A Área de Inspeção	224
C.2.4	A Área de Console	228

1. INTRODUÇÃO

Algum tempo atrás, começamos a receber notícias afirmando que viveríamos uma grande revolução tecnológica, mudando o modo como interagimos com o mundo e, principalmente, o modo como o mundo interage conosco. Essa revolução diz respeito ao surgimento da chamada Internet das Coisas (em inglês, *Internet of Things* - IoT), que prevê o surgimento de uma enorme rede, onde objetos físicos de nosso cotidiano podem estar conectados através da internet e aptos a se comunicarem entre si e com seus usuários, por meio de sensores e softwares inteligentes (ATZORI *et al.*, 2010).

A grande promessa da IoT é trazer mais conforto e produtividade para a população, permitindo que dispositivos eletrônicos quaisquer interajam entre si, provendo serviços diversos. Por exemplo, os sensores de presença de uma casa poderão, ao detectar a presença de uma pessoa, entrar em contato com o sistema de segurança da casa, que por sua vez contactaria o celular de todas as pessoas autorizadas a acessar a casa, e reconhecendo a localização de algum deles como sendo a localização do potencial intruso, poderia autenticar a presença de um usuário autorizado e ajudá-lo, abrindo o portão da garagem e acendendo as luzes do jardim. Caso contrário, poderia tomar outras medidas restritivas para tentar conter uma possível invasão não autorizada, ou reportar a tentativa de invasão. Para tanto, a IoT deve ser capaz de integrar uma quantidade extremamente grande de sistemas distribuídos e heterogêneos, onde muitos desafios técnicos e sociais ainda precisam ser investigados.

Um desses desafios diz respeito à capacidade de dois ou mais sistemas receberem, processarem, compreenderem e trocarem informações, um problema conhecido como interoperabilidade. De acordo com Martín *et al.* (2015) existem diferentes tipos de interoperabilidade: técnica, sintática, semântica e organizacional. A interoperabilidade técnica diz respeito aos protocolos e infraestrutura de hardware/software necessários para que dois ou mais sistemas possam operar entre si. Já a interoperabilidade sintática trata das questões de formatação sintática das mensagens transferidas pelos protocolos de comunicação entre estes sistemas. A interoperabilidade semântica, por sua vez, está no nível da aplicação, onde, por meio da linguagem, o significado das informações trocadas entre os sistemas e humanos deve ser corretamente interpretado para prover o serviço adequado. Por fim, a interoperabilidade organizacional diz respeito à capacidade das organizações de comunicarem e transferirem informações, mesmo usando uma variedade de sistemas diferentes e com infra-estruturas distintas. Por esse motivo, uma interoperabilidade organizacional depende de uma prévia interoperabilidade técnica, sintática e semântica.

Ainda, segundo Martín *et al.* (2015), relativo sucesso já vem sendo alcançado por diversos pesquisadores no que diz respeito às interoperabilidades técnica e sintática.

Entretanto, há pouco avanço nas questões envolvidas com a interoperabilidade semântica. Para tanto, uma prática utilizada por alguns pesquisadores atacando o problema da interoperabilidade semântica é o emprego de ontologias¹. Tal modelo permite representar explicitamente um domínio de conhecimento para os sistemas, de modo que estes consigam recuperar informações e gerar inferências a respeito do contexto abordado. Porém, dada a heterogeneidade dos sistemas envolvidos, é possível que dois sistemas diferentes acabem empregando diferentes ontologias para descrever uma mesma situação, o que acaba gerando incompatibilidade de comunicação entre eles. Para garantir a interoperabilidade semântica, deve-se integrar as diferentes ontologias através de um mapeamento que correlacione conceitos que representam um mesmo significado (ou significado similar), identificando semelhanças entre as entidades envolvidas. Mas essa tarefa não é tão simples como parece.

Dietze e Domingue (2008) afirmam que as ontologias, sendo uma abordagem simbólica, utilizam-se apenas de estruturas lógicas (ou seja, símbolos) para definir os conceitos que descrevem. Esse fato acaba acarretando no **problema da fundamentação do símbolo**, pois depende-se sempre de uma interpretação humana para conectar esses símbolos aos elementos do mundo a que se referem. Além do mais, diversos autores argumentam que uma visão computacional da linguagem como um processo baseado puramente na manipulação de símbolos, limita a capacidade e desempenho dos sistemas, uma vez que esse processo necessita de uma forma genuína de “significação”, que deve ser efetivamente adquirido e utilizado durante as interações do próprio sistema ((CLANCEY, 1997), (GLENBERG; ROBERTSON, 2000), (ZIEMKE, 2000), (ZIEMKE, 2001), (ROY; REITER, 2005), (PEZZULO *et al.*, 2011)).

Diante desse argumento fica claro que as abordagens puramente simbólicas impõem limitações severas para as pesquisas em sistemas inteligentes (BROOKS, 1990), mais especificamente, à maneira como esses sistemas representam seu conhecimento e o utilizam para requisitar serviços uns para os outros. De acordo com a abordagem da cognição incorporada e situada (*Embodied and Situated Cognition*) (THOMPSON; VARELA, 2001), o fenômeno da cognição é uma atividade experiencial, que ocorre durante a interação sensório-motora do sujeito com seu ambiente (MATURANA; VARELA, 2001), (ANDERSON, 2003). A inteligência que antes demandava símbolos e regras implantados por um projetista, agora é construída através de um processo dinâmico entre o sujeito e o mundo. Isso é uma quebra de paradigma, pois é o próprio sistema que constrói seu modelo mental de mundo, utilizado para criar e fundamentar suas representações simbólicas (JOHNSON-LAIRD, 1983), (JOHNSON-LAIRD; BYRNE, 2002).

¹ O termo ontologia pode ser empregado em diversas áreas de conhecimento, tal como na filosofia. Mas no contexto da ciência da computação uma ontologia pode ser vista como uma descrição formal (como uma especificação de um programa) dos conceitos e relacionamentos que eles estabelecem dentro de um domínio (GRUBER, 1993).

Atualmente existem diversas teorias na literatura demonstrando como a cognição incorporada e situada atua e como é aplicada em diversos segmentos de pesquisa (ver (LAKOFF; JOHNSON, 1999) e (CLARK, 1998)). No caso deste trabalho, estamos interessados em dotar os sistemas com a capacidade de comunicação através do emprego de sentenças composicionais, formadas por diferentes classes de palavras. Assim sendo, investigamos algumas teorias relacionadas com o surgimento da linguagem artificial, mais especificamente com seus aspectos gramaticais e sua fundamentação.

Analisando algumas referências, percebemos que aquelas que enfocam o aspecto léxico-semântico da linguagem (isto é, o uso de palavras isoladas e o seu significado) têm alcançado relativo sucesso, desenvolvendo modelos que atuam de forma semelhante aos seres vivos quando fazem uso dessa habilidade (ver, por exemplo, Hutchins e Hazlehurst (1995), Jung e Zelinsky (2000), Oliphant e Batali (1997), Oudeyer (1999), Steels (1999), Roy (2002), Vogt e Coumans (2002), Steels (2003), Loula (2004)). Entretanto, o mesmo sucesso não é observado nas referências que tentam explorar os aspectos gramaticais da linguagem (uso de sentenças compostas por palavras de diferentes classes, as denominadas sentenças composicionais - ver Batali (1998), Briscoe (1999), Kirby (2000), Chang (2008), Angluin e Becerra-Bonache (2008), Majewski e Zurada (2008)).

No nosso entendimento, a falta de sucesso dos trabalhos que simulam a emergência da linguagem gramatical seria proveniente de três considerações que fazemos a seguir.

1. A primeira diz respeito a uma recente descoberta da neurociência, (apresentada por Ardila (2012)), evidenciando que possuímos dois sistemas linguísticos independentes em nosso cérebro. Um deles é responsável pelas competências léxico-semânticas e outro é responsável pela competência gramatical. Até agora, a competência gramatical vem sendo tratada como uma evolução natural, e continuada do sistema léxico-semântico e não como um sistema a parte.
2. A segunda consideração diz respeito à utilização de teorias semióticas diádicas que não levam em questão a existência de signos naturais, tais como ícones e índices. A utilização de símbolos como a única forma de representação faz com que os sistemas atuais adotem de maneira implícita certas premissas que os tornam, automaticamente, limitados.
3. A terceira diz respeito à falta de um modelo coerente do que é o processo de percepção, pois é através dessa habilidade cognitiva que criamos nossas representações do mundo. Perceber não é ter sensações apenas, é ter sensações que compreendemos ((NOË, 2004)).

Na tentativa de superar as limitações apontadas por essas 3 considerações, propomos um modelo de percepção computacional que incorpora tanto a teoria dos sistemas de símbolos perceptuais proposta por Barsalou (1999), quanto a teoria dos espaços conceituais de Gärdenfors (2014a). Dessa forma, as informações sensorizadas por um sistema inteligente podem ser empregadas para segmentar espaços conceituais em regiões semânticas, remetendo a diferentes tipos de representações conceituais. Isto é, “símbolos modais” (de acordo com a nomenclatura de Barsalou) ou “ícones” (na nomenclatura de Peirce²) podem ser empregados para dar fundamentação a símbolos linguísticos. De acordo com Barsalou, símbolos linguísticos, por serem criados a partir de símbolos modais, são portanto fundamentados e seus significados dependem do reviver das experiências sensoriais que os originaram, um fenômeno que ele nomeou de **simulação mental**.

Portanto, neste trabalho, pretendemos trazer uma contribuição à área de linguagem gramatical artificial, apresentando um modelo formal para o mecanismo de simulação mental, capaz de atribuir significado a símbolos utilizados em sentenças gramaticais formadas por diferentes classes de palavras. Nosso pressuposto de trabalho parte da ideia de que para promover a criação de estruturas simbólicas composicionais é necessário criar uma etapa intermediária de representação, onde sinais dos sensores e atuadores serão traduzidos no que chamaremos mais adiante de **categorias ontológicas**. Essas categorias servirão para conceitualização dos objetos com os quais os sistemas interagirão e dos eventos estabelecidos, bem como para recuperar trechos das experiências que farão a fundamentação simbólica. Dessa forma, em contextos que envolvem linguagem, palavras de várias classes gramaticais, tais como substantivos, adjetivos, verbos e advérbios poderão surgir e ser empregados para estabelecer um sistema de comunicação.

De posse desse mecanismo de simulação mental, agentes artificiais participando de jogos de linguagem acabam por desenvolver um sistema semântico, compartilhado com seus interlocutores, sem depender previamente de seu projetista para lhes informar o significado dos símbolos trocados, garantindo assim a interoperabilidade semântica entre eles.

1.1 Hipótese de Pesquisa

A fundamentação do símbolo ganha destaque neste trabalho, pois ela diz respeito à maneira como os símbolos utilizados na linguagem estão correlacionados com as coisas do mundo que representam. A cognição incorporada e situada propõe uma nova abordagem para a fundamentação do símbolo, passando a considerar sinais oriundos de

² Na teoria semiótica do filósofo norte-americano Charles Sanders Peirce, *ícones* são signos que representam seu objeto pelo fato de possuírem, em si próprios, certas características ou propriedades dos objetos que representam

sensores e atuadores como representações mentais do mundo, a partir das quais a associação entre símbolos e a realidade pode ser estabelecida. Essa abordagem traz novos vislumbres para a área de emergência de linguagem artificial. Porém, analisando-a sob a luz da semiótica de Peirce, percebemos que simplificações são feitas e, que talvez, impediram avanços para a emergência de uma linguagem gramatical.

De acordo com a teoria semiótica de Peirce, diferentes tipos de representação do conhecimento podem ser unificados no conceito de signo. Segundo Peirce, esses signos podem ser de diferentes classes. Dessa forma, tanto sinais sensório-motores, quanto símbolos, são signos, só que de classes distintas. Portanto, o símbolo não é um elemento universal de representação, mas o ápice de uma sequência de processos de interpretação, que começa com signos mais elementares, como ícones e índices, e por meio de uma sequência sucessiva de interpretações pode emergir a partir destes.

Uma pergunta que surge nesse contexto é: será que é possível fazer uma associação direta entre símbolos e elementos da representação sensório-motora? Segundo os relatos da literatura que envolvem a emergência de significado para palavras isoladas (sistema léxico-semântico), isso seria possível. Entretanto, cabe ressaltar as evidências de [Ardila \(2011\)](#), de que possuímos dois sistemas linguísticos independentes. Aparentemente, o sistema léxico-semântico está modelado de maneira adequada, mas ainda não temos um modelo satisfatório para o sistema gramatical. Desta forma, poderiam haver casos em que uma interpretação direta dos símbolos na representação sensório-motora não seria factível. E qual seria a solução para este problema?

Segundo [Gudwin \(2012\)](#), para que símbolos composicionais (sentenças com palavras pertencentes a diferentes classes) sejam traduzidos em elementos da representação sensório-motora, é necessária uma tradução intermediária em representações que chamamos aqui de *categorias ontológicas*. Trata-se de um modelo de representação da realidade que mapeia as regiões semânticas de um espaço conceitual em categorias de diversas naturezas. E essas categorias são capazes de representar muito mais do que apenas objetos, suas propriedades e partes constituintes, mas também a maneira como eles se relacionam e se transformam seguindo uma trajetória temporal.

Sendo assim, a nossa hipótese é fundamentada no seguinte pressuposto: **para que haja a criação de estruturas simbólicas composicionais, deve existir um tipo de representação intermediária que, durante o processo de categorização, segmente a realidade em termos de categorias ontológicas.** Inicialmente, os sensores e atuadores fariam uma primeira tradução da realidade, gerando signos da representação sensório-motora (sinais sensoriais e motores). Em seguida, os signos da representação sensório-motora seriam traduzidos em signos que compõem diferentes categorias. Por fim, essas categorias poderiam ser traduzidas em termos linguísticos, gerando signos da representação simbólica, incluindo substantivos, verbos, adjetivos, advérbios,

sentenças e pequenos textos sobre um objeto. Seguindo a trajetória inversa, teremos a interpretação dos símbolos. Para isso, sentenças seriam inicialmente traduzidas em signos das categorias ontológicas. Posteriormente, esses signos seriam traduzidos em termos da representação sensório-motora, que estariam em uma relação direta com a realidade por meio de sensores e atuadores. Essa proposta pode ser visualizada na Figura 1.

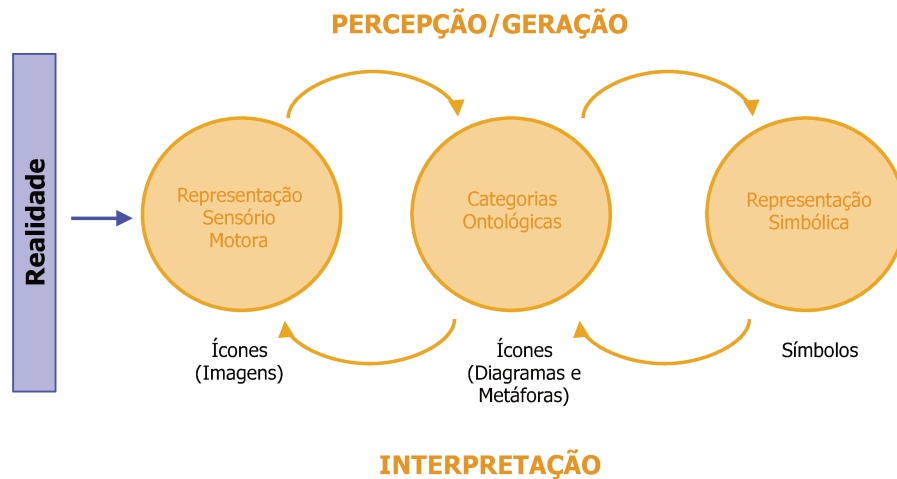


Figura 1 – Hipótese proposta para a fundamentação de símbolos gramaticais

Ao contrário dos signos da representação sensório-motora, que representam a realidade em termos dos sinais que podem ser mensurados por sensores e atuadores, a representação oriunda das categorias ontológicas é capaz de descrever o mundo como uma coleção de objetos, que podem ter seus atributos modificados e suas *affordances*³ executadas durante o transcorrer do tempo.

Vale ressaltar que os signos que compõem as categorias ontológicas nada mais são do que uma tradução dos signos da representação sensório-motora em um nível de abstração mais alto, e todos eles seriam ícones, porém classificados como imagens, diagramas ou metáforas. Peirce define uma imagem como um ícone que possui em si mesmo, os mesmos atributos de seu objeto. Quando capturamos um pequeno fragmento instantâneo da realidade (um *frame*, em nossa terminologia), na forma de uma coleção de propriedades físicas capturadas por sensores e/ou comandos de atuação de atuadores, estamos diante de uma *imagem*. Ainda, de acordo com Peirce, um *diagrama* seria um signo que representa por meio das relações entre suas próprias partes, relações análogas que ocorrem em seu objeto. Desta forma, quando representamos espacialmente uma sequência de imagens (sequências de *frames*, formando o que chamamos de caminhos, ou *paths*⁴, em

³ O termo *affordance* foi introduzido por Gibson (1986) para indicar os convites à ação que os diferentes objetos, dependendo de sua categoria, fazem aos agentes. Essa terminologia será melhor explorada e explicada nos capítulos 2 e 5 deste trabalho

⁴ Um *path* aqui pode ser entendido como um fragmento espaço-temporal da realidade, composto por um conjunto de objetos descrevendo uma determinada trajetória no tempo, com relação a seu espaço de atributos.

nossa terminologia), coletadas sequencialmente no tempo, podemos entender que a relação espacial entre cada duas imagens na sequência corresponde à relação temporal real que há entre os valores das propriedades, modificando-se no tempo. Estamos portanto, diante de um diagrama, segundo Peirce. Por fim, Peirce ainda nos apresenta ao que ele chama de *metáfora*, segundo ele, um signo que representa por meio de algum outro tipo de paralelismo estabelecido com seu objeto. Em nossa interpretação, objetos computacionais, do tipo utilizado em linguagens orientadas a objetos são exatamente metáforas no sentido Peirceano. Isso porque, **objetos computacionais**, enquanto signos, não apresentam, em si mesmos, os atributos de seus objetos⁵ (ou seja, não são imagens), nem apresentam na relação de suas partes, possíveis relações entre partes de objetos (ou seja, também não são diagramas), mas apresentam uma similaridade numérica entre os valores dos atributos físicos de seus objetos e os números armazenados internamente em seus atributos. Dessa forma, existe uma similaridade numérica entre os atributos dos objetos computacionais e os atributos dos objetos do signo. Com isso, podemos concluir que os objetos computacionais são *metáforas* de seus objetos reais. Esses objetos computacionais, com seus atributos e métodos, bem como sua transformação temporal, compondo eventos, episódios e cenas são a base para a construção do que estamos chamando aqui de categorias ontológicas. Posteriormente, esses objetos estarão relacionados a símbolos quaisquer, que quando organizados linearmente na forma de sentenças e frases, poderão ter seu significado explicitado na trajetória de um conjunto de objetos no tempo, permitindo a associação do fragmento de realidade espaço-temporal com uma sentença que descreve o que acontece nesse fragmento de espaço-tempo, sendo a base para nosso processo de fundamentação simbólica. Maiores detalhes sobre esse processo são apresentados no capítulo 5, mais especificamente, na seção 5.2.

1.2 Objetivos

O objetivo principal deste trabalho é apresentar um modelo formal para o mecanismo de simulação mental, que aqui será empregado para tratar questões da emergência da linguagem gramatical artificial. Por esse motivo ele também contará com os seguintes objetivos específicos:

1. Desenvolver a infraestrutura necessária para criar e interpretar sentenças composicionais.
2. Estabelecer meios para criação de um sistema semântico compartilhado para gerar interoperabilidade semântica entre falantes de uma língua.

⁵ Objetos aqui, no sentido de objetos do signo. Cuidado para não confundir aqui um objeto computacional com o objeto de um signo. São conceitos diferentes que têm uma terminologia semelhante, que pode levar à confusão.

3. Permitir que sistemas inteligentes, ao empregarem o modelo proposto de simulação mental, possam interagir e se coordenar através de atos comunicativos.

1.3 Metodologia de Trabalho

Neste trabalho de pesquisa foi desenvolvida uma implementação computacional para o mecanismo de simulação mental proposto por Barsalou (1999) que emprega como *framework* semântico espaços conceituais. Ele foi aplicado em questões relacionadas à emergência de linguagem gramatical artificial. Por esse motivo três tipos de experimentos foram propostos para demonstrar sua viabilidade e desempenho. São eles: (i) a segmentação de um espaço conceitual e a criação, em tempo de execução, das categorias ontológicas usadas para fundamentar símbolos linguísticos; (ii) a funcionalidade de cada módulo que compõe o mecanismo do modelo proposto de simulação mental; (iii) o emprego do modelo em três tipos distintos de jogos de linguagem estabelecidos por uma comunidade de agentes.

1.4 Lista de Publicações

Os resultados apurados a partir desse trabalho de pesquisa geraram até o presente momento as seguintes publicações:

- **Trabalhos publicados em periódicos:**

- GUDWIN, Ricardo et al. The multipurpose enhanced cognitive architecture (MECA). *Biologically Inspired Cognitive Architectures*, v. 22, p. 20-34, 2017.
- DE PAULA, Suelen M.; GUDWIN, Ricardo R. Evolving conceptual spaces for symbol grounding in language games. *Biologically Inspired Cognitive Architectures*, v. 14, p. 73-85, 2015.

- **Trabalho publicado em evento:**

- DE PAULA, Suelen Mapa; CARVALHO, Micael Cabrera; GUDWIN, Ricardo Ribeiro. Building up Conceptual Spaces: An ESOM Supported Strategy. In: 2015 Brazilian Conference on Intelligent Systems (BRACIS). IEEE, 2015. p. 122-127.

1.5 Organização do Trabalho

Como forma de auxiliar a compreensão deste trabalho, adotou-se a seguinte organização:

- No presente capítulo: **Introdução** - apresentamos a introdução para este trabalho, nossa hipótese de pesquisa, nossos objetivos, a lista de publicações e a organização que adotamos na elaboração do trabalho.
- No capítulo 2: **Fundamentação Teórica** - apresentaremos os trabalhos das ciências cognitivas que serviram de *background* e inspiração para a construção do mecanismo de simulação mental descrito no capítulo 5;
- No capítulo 3: **Jogos de Linguagem** - apresentaremos a definição para *jogos de linguagem* que serviram de inspiração para implementar os atos comunicativos do capítulo 6;
- No capítulo 4: **Trabalhos Correlatos** - serão analisados os trabalhos que, assim como este, abordam a emergência de linguagem gramatical sob a perspectiva das simulações mentais, espaços conceituais e gramática de construções, focando, principalmente, nos seus pontos fortes e críticas;
- No capítulo 5: **Mecanismo Computacional para Simulação Mental** - apresentaremos o modelo de um mecanismo de simulação mental proposto por este trabalho;
- No capítulo 6: **Mecanismo de Simulação Mental: Modo Linguagem** - apresentaremos um pacote de linguagem que irá fornecer insumos para validar a capacidade de fundamentação simbólica do mecanismo de simulação mental;
- No capítulo 7: **Experimentos Computacionais** - apresentaremos os protocolos de jogos de linguagem que foram escritos para validar a implementação computacional do mecanismo de simulação mental;
- No capítulo 8: **Conclusão** - apresentaremos as principais conclusões apuradas na condução deste trabalho, bem como as perspectivas de trabalhos futuros.

2. FUNDAMENTAÇÃO TEÓRICA

Pensar como símbolos linguísticos são gerados e ganham significado nos remete às origens do conhecimento que eles comunicam. Isto é, de onde ele veio? Ou como ele é registrado na mente? Essas questões estão relacionadas a um tópico da Inteligência Artificial (IA) conhecido como **representação do conhecimento**.

O problema da representação do conhecimento diz respeito à maneira como um sistema inteligente registra e manipula informações acerca do mundo para ser capaz de extrair conhecimento (DAVIS *et al.*, 1993; SOWA *et al.*, 2000). A combinação de informações e regras compõe a base de conhecimento do sistema, que pode ser empregada para gerar inferências acerca deste contexto. No período entre 1980 e 1990, tais premissas para a construção de sistemas inteligentes tornaram-se bastante populares, e o conceito de inteligência passou a ser entendido como a capacidade de um sistema manipular representações abstratas por meio de regras, a fim de gerar conhecimento (ANDERSON, 2003). Essa tendência ficou conhecida como simbolismo, demarcando o período da IA clássica.

Segundo Sowa *et al.* (2000), para que um sistema consiga executar inferências sobre sua base de conhecimento, as informações e regras que a compõem devem ser descritas de maneira genérica e abstrata, demandando de formalismos específicos para tal. Existe na literatura um grande número desses recursos, tais como árvores de decisão, regras de produção, regras lógicas, redes semânticas, *frames*, procedimentos e outros. Em sua essência, a maioria desses utiliza representações simbólicas. Porém, como argumentaremos ao longo deste trabalho, representações puramente simbólicas acabam trazendo limitações para o desenvolvimento de sistemas inteligentes, como veremos mais adiante.

Neste capítulo, apresentamos alguns formalismos que se contrapõem a essa prevalência do simbolismo, complementando os formalismos mais convencionais, e permitindo a modelagem de outros tipos de representações, não simbólicas, como por exemplo, as representações icônicas e as representações indexicais. Assim sendo, nas seções que se seguem falaremos sobre ontologias, o problema da fundamentação do símbolo, a teoria semiótica de Peirce, o sistema de símbolos perceptuais de Barsalou e a teoria dos espaços conceituais, de Gärdenfors. Esses foram os referenciais teóricos que fundamentaram a proposição do modelo proposto para o mecanismo de simulação mental apresentado no capítulo 5.

2.1 Ontologias

No sentido filosófico, o termo ontologia diz respeito a “aquilo que é, há ou existe”. Seu objetivo é classificar as entidades da realidade, definindo seu vocabulário e as formulações canônicas a respeito delas (FRANK, 2003). Diferentes filósofos, no passado (e.g. Aristóteles, Kant e Peirce) propuseram diferentes repertórios de categorias ontológicas, por meio das quais seria possível segmentar a realidade. Dessa forma, um modelo da realidade pode ser definido a partir de um conjunto de categorias ontológicas e de como os elementos dessas categorias interagem entre si.

Na área da ciência da computação, o interesse por ontologias surgiu quando os projetistas de sistemas inteligentes perceberam que diferentes desenvolvedores de bases de conhecimento utilizavam as vezes de termos bastante diferentes para se referir a um mesmo conhecimento, criando incompatibilidades entre tais sistemas, e portanto de nada valia a preocupação com formalismos de representação do conhecimento (regras, *frames*, redes semânticas, procedimentos e etc) se não existisse algum tipo de padronização sobre o conhecimento do domínio em que se deseja trabalhar (CHANDRASEKARAN *et al.*, 1999).

Assim, surge o conceito de ontologia, que segundo Gruber (1993) pode ser visto como uma especificação explícita de um repertório de conceitos. Uma ontologia visa representar tudo que existe em um domínio, sendo a mesma composta por um vocabulário representacional com definições precisas para seus termos, mais um conjunto formal de axiomas escritos em linguagem natural, que permite realizar inferências sobre ele.

Existem, de fato, várias definições de ontologias. Porém, qualquer que seja sua definição, a construção de uma ontologia sempre resultará em uma descrição formal dos conceitos de um domínio, que representará o entendimento consensual de um grupo de pessoas que atuam sobre o mesmo domínio.

Por dispor de uma base de conceitos sobre um domínio de conhecimento, as ontologias são empregadas em diferentes áreas de atuação, inclusive naquelas que envolvem interoperabilidade semântica, tal como na comunicação entre dois sistemas inteligentes. Porém, como aponta Dietze e Domingue (2008), o uso de ontologias em cenários que demandam comunicação e linguagem acaba gerando mais problemas do que soluções, por exemplo:

1. Por se tratar de uma abordagem simbólica, ontologias demandam de formalismos de representação simbólicos para definir os conceitos empregados em sua conceituação. Esse fato acaba acarretando no problema da fundamentação do símbolo. Harnad (1990) afirma que, para existir formas genuínas de significação, os símbolos não podem ser descritos por meio, apenas, de suas relações com outros símbolos.

2. Além disso, a comunicação entre sistemas distintos pode acarretar no emprego de ontologias distintas para descrever a mesma realidade. Mesmo que as ontologias sejam semelhantes, dificilmente elas se igualam a outras, pois a noção de similaridade não é algo implícito a elas. Apesar do emprego de ferramentas adicionais para realizar a identificação manual ou semi-automática dessas relações de similaridade, tal atividade é dispendiosa e requer muito esforço da parte de seus projetistas.

Diante desse argumentos, fica claro que, em contextos onde há emergência de linguagem, o emprego de ontologias não é tão vantajoso, dependendo de recursos adicionais para contornar os problemas apontados.

2.2 O Problema da Fundamentação do Símbolo

Segundo [Balkenius et al. \(2000\)](#), a manipulação de símbolos é requisito fundamental para aquisição de linguagem. Entretanto, gerar símbolos a partir de regras formais definidas por um projetista está longe de representar um modelo razoável de como os humanos adquirem e usam a linguagem, o que acaba implicando em um problema conhecido na literatura como o **problema da fundamentação do símbolo**.

Segundo [Harnad \(1990\)](#), o problema da fundamentação do símbolo diz respeito à maneira como a interpretação semântica gerada por sistemas simbólicos formais pode ser feita intrinsecamente, sem depender de uma interpretação externa feita por uma mente humana. Essa definição abre espaço para o seguinte questionamento: como o significado de símbolos manipulados unicamente por meio de suas formas arbitrárias pode ser fundamentado a partir de outros símbolos, que são igualmente sem sentido? De acordo com [Searle et al. \(1980\)](#), em seu famoso argumento do Quarto Chinês, este problema seria análogo ao que aconteceria com um indivíduo que, do interior de um quarto isolado, recebe perguntas redigidas em chinês, e gera respostas também em chinês mas, sem de fato compreender o idioma chinês, somente aplicando regras sintáticas aos símbolos recebidos, produzindo respostas de maneira mecânica.

Diversos autores argumentam que uma visão computacional da linguagem como um processo fundamentalmente sintático, baseado na manipulação de símbolos, limita a capacidade e desempenho de sistemas artificiais, uma vez que esse processo necessita de “fundamentação” e de uma forma genuína de “significação”, que deve ser efetivamente adquirida e utilizada durante as interações do sistema ([CLANCEY, 1997](#)), ([GLENBERG; ROBERTSON, 2000](#)), ([ZIEMKE, 2000](#)), ([ZIEMKE, 2001](#)), ([ROY; REITER, 2005](#)), ([PEZZULO et al., 2011](#)).

Diante desses argumentos, fica claro que a IA clássica impõe essa limitação para as pesquisas em sistemas inteligentes ([BROOKS, 1990](#)). Isso motivou a criação de

outras abordagens que pudessem oferecer mais coerência e flexibilidade para a construção de sistemas inteligentes, como é o caso do conexionismo (RUMELHART; MCCLELLAND, 1986) e da cognição incorporada e situada (*Embodied and Situated Cognition*) (THOMPSON; VARELA, 2001). Segundo Harnad (1990), ambas as abordagens são promissoras porque propõem mecanismos *bottom-up* (tais como redes neurais, sistemas fuzzy, espaços conceituais, simulações mentais e outros mais) para emergência de representações não-simbólicas, que podem ser usadas para fundamentar representações simbólicas.

Na visão de Harnad, as representações não simbólicas podem ser de dois tipos: (1) **representações icônicas**, que podem ser vistas como entidades análogas às *projeções sensoriais*¹ de eventos e objetos; e (2) **representações categóricas**, que são detectores de características aprendidas ou inatas e servem para identificar objetos e eventos a partir de suas projeções sensoriais.

Para evitar o problema da fundamentação do símbolo, as representações icônicas ou categóricas devem ser empregadas para associar símbolos elementares a representações simbólicas de alta-ordem². Dessa maneira, representações simbólicas de alta-ordem, baseadas nesses símbolos elementares, consistem em cadeias de símbolos que descrevem relações de associações, como por exemplo, X é um Y que é Z ou como mostra a Figura 2. (HARNAD, 1990).

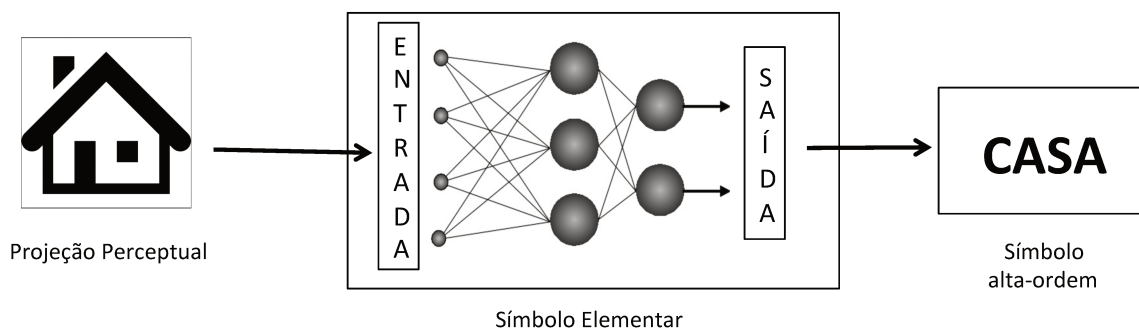


Figura 2 – Relação de associação entre representações simbólicas

Ao sensorar a imagem de uma casa, uma rede neural usa essas informações para configurar seus pesos e, assim, aprender a identificar o que pode ser uma casa. A configuração da rede (aqui chamada de símbolo elementar) é associada ao símbolo de alta-ordem “casa”. Portanto, tanto a projeção perceptual da imagem, quanto a configuração dos pesos da rede neural também representam a palavra “casa”.

Analisando o problema da fundamentação do símbolo à luz de pesquisas mais

¹ Entende-se que eventos e objetos podem ser conhecidos por meio de sua projeção em um espaço de propriedades, onde pode-se amostrar essas propriedades por meio de sensores

² Neste trabalho, utilizaremos o termo “símbolo de alta-ordem”, de maneira análoga a Harnad (1990), para designar aquilo que na terminologia Peirceana seria chamado simplesmente de “símbolo”, e que na terminologia de Barsalou (1999) é chamado de símbolo amodal.

recentes percebemos que a comunidade científica se divide entre duas posições distintas. Há aqueles pesquisadores, como [Steels \(2008\)](#), que acreditam que esse problema já foi resolvido. Na concepção de [Steels \(2008\)](#) o problema da fundamentação dos símbolos já foi resolvido, pois atualmente entendemos o suficiente para criar experimentos nos quais grupos de agentes auto-organizam sistemas simbólicos que são fundamentados a partir de suas interações sensoriomotoras com o mundo, sem um projeto prévio para fornecer os símbolos ou seu significado.

Por outro lado, outros pesquisadores afirmam o contrário. Por exemplo, [Taddeo e Floridi \(2005\)](#), após realizarem uma pesquisa sobre 8 estratégias desenvolvidas para resolver o problema da fundamentação do símbolo concluíram que nenhuma delas forneceu uma solução válida para o problema, que portanto permanece um problema em aberto. Eles ainda afirmam que ele só terá uma solução viável quando a capacidade semântica para gerar representações for projetada em conjunto com os aspectos da sua fundamentação.

O que podemos entender sobre esse debate é que ainda há muito o que se aprender sobre este problema de pesquisa. E que as respostas surgirão a partir de experimentos com sistemas artificiais, através dos quais possamos entender progressivamente os aspectos da interação dinâmica entre processos de conceitualização, significado e símbolos.

2.3 A Teoria Semiótica de Peirce

Se a representação do conhecimento por meio de formalismos simbólicos é um problema, por que ao invés de nos preocuparmos com ela diretamente, não nos concentramos em definir o que seja representação e suas modalidades, e como elas podem representar outras coisas? Esta questão é amplamente tratada pela Semiótica – a doutrina dos signos (CP 2.227)³ que estuda o fenômeno da significação. Dizendo de outra forma, a semiótica é a ciência que estuda a maneira como os indivíduos (sejam eles naturais ou artificiais) dão significado a tudo aquilo que os cerca. Embora o processo de significação já tenha sido abordado por outros filósofos e linguistas do passado, foi Charles Sanders Peirce (1839-1914) quem definiu e estruturou a semiótica como uma área de pesquisa independente.

Ao longo de seus estudos, Peirce chegou a um conjunto de três categorias fundamentais: **primeiridade** (algo que é como é independente de qualquer outra coisa), **secundidade** (algo que só é o que é, em relação a um outro) e **terceiridade** (algo que está no meio ligando um primeiro a um segundo) (CP 1.347). O princípio que envolve essas categorias é bem abstrato. Entretanto isso foi feito de forma proposital para que

³ O trabalho de Charles S. Peirce é citado como CP seguido do volume e parágrafo referindo-se aos *Collected Papers* ([PEIRCE, 1958](#))

elas conseguissem denotar coisas diversas e distintas. Elas resumem em si a essência de sua teoria e é a partir delas que surge a classificação dos signos. Mas antes de entrarmos nos detalhes dessa classificação, vamos entender o que é um signo.

O termo signo é fundamental na teoria de Peirce. Segundo [Santaella \(2000\)](#), é possível encontrar dezenas de definições distintas para ele. No entanto, na maioria delas é predominante a relação entre ele (o signo), o objeto que ele representa e seu interpretante, como pode ser observado nesta citação:

Defino um signo como qualquer coisa que, de um lado, é assim determinado por um objeto e, de outro, assim determina a ideia na mente de uma pessoa, esta última determinação, que eu denomino de interpretante do signo, é, desse modo, mediatamente determinada por aquele objeto. Um signo assim tem relação triádica com seu objeto e com seu interpretante (CP 8.343).

A partir dessa definição, podemos entender que o signo é algo que representa um objeto. É também ele quem determina o interpretante, aquilo que ocorre na mente, sendo que esse interpretante também está associado ao mesmo objeto. Assim, o signo está em uma relação triádica com seu objeto e seu interpretante, compondo uma tríade no processo de representação. Essa relação é denominada por Peirce de semiose (CP 5.484), e é durante a semiose que o signo entra em ação, ocorrendo o processo de significação. Neste caso, o signo levará à mente do sujeito algo exterior, que é o objeto, e permanecerá em seu lugar, transmitindo o seu significado, gerando assim o interpretante ([SANTAELLA, 2000](#)).

De acordo com a definição de semiose, para que um signo possa existir, ele sempre deve estar correlacionado com um objeto e um interpretante, enquadrando-se na categoria de uma terceiridade. A classificação dos signos é derivada a partir do próprio modelo de signo, que é aplicado recursivamente para dar origem a outros tipos de signos. Sendo assim, é possível identificar três tipos de relação: o signo em relação a ele mesmo (signo-signo), o signo em relação ao objeto (signo-objeto) e o signo em relação ao interpretante (signo-interpretante). Para o caso deste trabalho, nós iremos nos restringir à segunda relação (signo-objeto), pois dela foi originada o que Peirce denominou como sendo a mais fundamental divisão dos signos (CP 3.275).

A aplicação das categorias de primeiridade, secundidade e terceiridade à relação signo-objeto deu origem a três classes distintas de signos – ícones, índices e o símbolos.

Os dois primeiros signos, ícone e índice, são chamados de signos naturais. Eles são denominados assim, porque representam o objeto naturalmente, independente de uma convenção pré-estabelecida. Já o símbolo, não é visto como um signo natural, pois para representar um objeto, ele demanda uma convenção prévia, arbitrária, fruto de um hábito do pensamento.

No caso da relação signo-objeto ser uma primeiridade, tem-se um ícone. Este é um tipo de signo que representa seu objeto por possuir algum tipo de similaridade com o mesmo. É o caso de uma fotografia, gravura ou diagrama que, de alguma forma, são similares àquilo que representam. Observe que essa similaridade pode ocorrer de diversas maneiras diferentes. No caso da relação signo-objeto ser uma secundidade, tem-se um índice. Este é um tipo de signo que representa seu objeto por direcionar a atenção do intérprete para ele. Isso pode ocorrer de diversas maneiras, por exemplo, por meio da existência de uma relação física, causal ou espaço-temporal que permita a ele direcionar a atenção para seu objeto. É o que acontece quando utilizamos uma seta para indicar algo, ou quando vemos fumaça e a associamos à existência de fogo. Por último, quando a relação signo-objeto é uma terceiridade, tem-se um símbolo. Este é um tipo de signo que representa seu objeto por força de uma conexão arbitrária com o mesmo, estabelecida por força de uma lei, convenção, regra ou hábito que cria esta ligação. É o que acontece quando nomeamos alguma coisa. Por exemplo, a palavra “mesa” é totalmente arbitrária com relação ao objeto [mesa]. Tanto é que em diferentes línguas, diferentes nomes podem ser atribuídos ao objeto [mesa]. Dessa forma, a conexão entre o símbolo “mesa” e o objeto [mesa] depende de incorporarmos uma regra (ou lei, ou hábito), que é aprendida quando começamos a ganhar uma competência linguística.

Se aplicarmos novamente as três categorias de primeiridade, secundidade e terceiridade a ícones, índices e símbolos, podemos encontrar sub-tipos destes. Por exemplo, se analisarmos um ícone sob a perspectiva de primeiridade, secundidade e terceiridade, encontraremos três sub-tipos deste: imagens, diagramas e metáforas. Uma imagem é um tipo de ícone que representa seu objeto por apresentar em si mesmo certas qualidades dele (primeiridade). Um diagrama é um ícone que representa seu objeto porque as mesmas relações que existem em suas partes podem ser encontradas nas partes de seu objeto (secundidade). E uma metáfora é um ícone que representa seu objeto por possuir algum outro tipo de similaridade com seu objeto, que não seja possuir em si mesmo as qualidades do objeto, nem possuir em si mesmo relações existenciais com ele, mas por conta de compartilhar com ele algum tipo de lei, regra ou hábito (terceiridade), em que ambos são classificados de maneira análoga (ou seja, é possível estabelecer algum tipo de analogia entre eles).

A semiótica de Peirce é uma teoria diferenciada, porque prevê uma grande variedade de tipos de signos. Segundo [Ardila \(2011\)](#) uma das condições para que a linguagem gramatical possa ocorrer é a existência de diferentes classes de palavras. O fato de nos basearmos numa teoria semiótica que contempla diferentes tipos de signos, nos oferece mais possibilidades para representar detalhes do objeto em questão e, assim, poder abstraí-los em símbolos que representam diferentes classes de palavras. Dessa forma, além de substantivos, podemos mapear também verbos, advérbios, adjetivos e outros.

2.4 A Teoria do Sistema de Símbolos Perceptuais de Barsalou

Quando se fala de representação icônica, um processo cognitivo que se destaca é o da percepção. É a partir dela que criamos representações mentais icônicas e, posteriormente, podemos utilizá-las para alguma finalidade cognitiva (CHALMERS *et al.*, 1992). Entretanto, uma dúvida que subsiste é como representações significativas podem ser formadas a partir da percepção. Para tentar responder essa questão, a percepção deve ser vista como um processo cognitivo que vai além de perceber coisas. Ela deve atuar como um sistema conceitual, que desenvolve estruturas capazes de representar entidades e eventos da experiência, a partir das quais é possível fazer inferências para extrair informações (BARSALOU, 1999).

Algumas abordagens atuais, assim como aquelas da IA clássica, entendem a percepção como um processo responsável por capturar as informações do ambiente e repassá-las para outros processos cognitivos. A percepção, vista dessa forma, atua como um sistema de registro que captura informações e cria cópias, como em uma fotografia. Mas segundo Noë (2004), a percepção é muito mais do que isso. É a capacidade de gerar entendimento sobre aquilo que percebemos. Chalmers *et al.* (1992) também corroboram com esta ideia, ao afirmarem que existem dois processos subjacentes à percepção, um de baixo nível, que é responsável pela sensibilização e outro de alto nível, que é responsável pela extração de significado daquilo que foi percebido.

Quando um sistema perceptual atua como um sistema de registro, nos deparamos com o problema da fundamentação do símbolo. Barsalou (1999) afirma que esse problema ocorre porque as representações mentais (no sentido do cognitivismo) criadas atuam como **símbolos amodais**⁴. Isso quer dizer que esses símbolos não correspondem aos estados perceptuais que os produziram e a ligação entre eles (símbolos e estados perceptuais) é arbitrária.

Como forma de tratar esse problema, Barsalou (1999) desenvolveu uma teoria que denominou de sistema de símbolos perceptuais (*PSS - Perceptual Symbol System*). De acordo com esta teoria, um sistema de símbolos perceptuais é aquele capaz de integrar diferentes experiências que descrevem um objeto ou situação em uma estrutura denominada **simulador**, para dar origem a uma categoria. Além disso, inferências sobre categorias podem ser feitas a partir do reviver mental de trechos do conteúdo desse simulador, um fenômeno que Barsalou (1999) denominou de **simulação mental**. Portanto, é a simulação mental que permite a criação de um sistema perceptual simbólico, o que é

⁴ Vale ressaltar aqui que a definição de símbolo para Barsalou difere da definição de símbolo de Peirce. Aparentemente, Barsalou (1999) desconhecia a teoria semiótica Peirceana, e portanto não pode se valer de sua terminologia. Barsalou utiliza o termo “símbolo” de maneira equivalente a “signo” na teoria Peirceana. O que Barsalou (1999) chama de “símbolo modal” seria um “ícone” na terminologia Peirceana.

feito por meio de duas vias. Uma, responsável por criar os simuladores que darão origem às categorias, e outra responsável por criar simulações, a partir das quais inferências sobre uma categoria podem ser feitas.

Durante uma *experiência*⁵, estados perceptuais surgem a partir do sistema sensório-motor. Alguns desses estados são extraídos via atenção seletiva e armazenados permanentemente na memória de longo prazo, sendo denominados como **símbolos modais**⁶. Os símbolos modais, ao contrário dos amodais, servem como referentes do mundo, pois possuem semelhança estrutural com os estados perceptuais que representam. São construídos sequencialmente, componente por componente, a partir de cada experiência vivenciada. Além do mais, são vistos como entidades multimodais, pois são originados a partir de estados perceptuais de diferentes órgãos sensoriais, inclusive da introspecção.

Uma vez que os símbolos modais vão para a memória de longo prazo, eles passam a se referenciar mutuamente, criando uma teia representacional, onde passam a depender uns dos outros. São organizados ao longo de um simulador, que funciona como uma categoria ou um conceito. Considere, por exemplo, um simulador que representa o conceito de um carro. Ao longo dele, podemos encontrar diferentes características do carro, tais como: informações visuais, informações auditivas, informações somatosensoriais e outras. O resultado é uma estrutura que acumula o conteúdo modal para a categoria carro. Neste contexto, o conceito do carro deixa de ser algo isolado e passa a ser situado em cenários específicos, o que Barsalou (1999) denominou de conceitualização situada. Portanto, à medida em que diferentes simuladores são criados, um sistema conceitual é formado.

Visto que o simulador tem em si o conteúdo modal de uma categoria, ele pode ser empregado tanto para representá-la, quanto para interpretá-la. Para isso, basta fazer com que pequenos trechos de seu conteúdo sejam experimentados novamente, através de uma simulação – algo análogo ao que ocorre durante a imaginação. Assim, ao reviver traços da experiência que deu origem a uma categoria, um indivíduo é capaz de fazer inferências sobre ela, permitindo que ele se recorde não apenas do fato em si, mas também de outras informações que são subjacentes à experiência, tal como todas as sensações que ocorreram no momento real em que ela se concretizava. Essa simulação complementa o sistema conceitual que foi formado, fornecendo meios de se chegar ao significado das categorias que ele representa, finalizando assim o processo que dá origem à simulação mental.

Barsalou (1999) vai além, ao afirmar que a partir de um sistema perceptual

⁵ No contexto da filosofia da mente, experienciar o mundo corresponde ao fenômeno subjetivo de estar em contato com o mundo, incluindo não somente seu sensoreamento, mas também as emoções, vontades e desejos, além de possivelmente outros estados mentais subjacentes a esse contato.

⁶ De acordo com a definição de símbolo proposta por Peirce e que foi assumida por este trabalho, o que Barsalou (1999) chama de símbolo modal, entendemos como sendo um ícone.

simbólico é possível obter três propriedades que no nosso entendimento são fundamentais para o surgimento de linguagem. A primeira delas é a **produtividade**. Ela diz respeito à habilidade de construir outras categorias a partir da combinação de categorias já existentes. Já a segunda propriedade é a criação de **proposições**. Ela diz respeito à habilidade de descrever um objeto ou situação a partir do conhecimento obtido durante a simulação mental. E por fim, temos a terceira propriedade, que é a criação de **conceitos abstratos**. Essa propriedade diz respeito à habilidade de descrever objetos ou eventos de uma experiência a partir daquilo que está representado em seu simulador, ou seja, é criar uma categoria.

Fazendo uma analogia dessas três propriedades com a linguagem, a produtividade pode ser vista como a combinação de diferentes símbolos para dar origem a outros símbolos, quando, por exemplo, criamos palavras. Já a criação de proposições é o entendimento que extraímos dos símbolos quando, por exemplo, lemos um texto ou conversamos com alguém. E a criação de conceitos abstratos é o mesmo que criar os símbolos que, neste caso, podem ser uma palavra ou frase, para descrever uma entidade ou contexto. Portanto, não nos restam dúvidas de que a teoria do sistema de símbolos perceptuais de Barsalou (1999) é uma proposta importante para o funcionamento da percepção e que pode ser aplicada ao contexto deste trabalho.

2.5 A Teoria do Espaço Conceitual

Um problema inerente ao contexto da linguagem é como as palavras ganham significado e como seu significado pode ser representado na mente de um indivíduo sem depender de um especialista. Para lidar com este problema, Gärdenfors (2004) introduz sua teoria dos espaços conceituais como uma alternativa .

A hipótese de Gärdenfors (2004) é que nossa mente organiza a informação envolvida durante a comunicação (tais como, palavras, sentenças, objetos, eventos e ações) em um formato que pode ser modelado em termos geométricos, a partir do que ele chamou de espaços conceituais. Enquanto atos comunicativos se desenrolam, um sistema semântico é desenvolvido e torna-se compartilhado entre seus interlocutores. Assim, símbolos podem ser aprendidos e compreendidos, pois estão correlacionados ao seu significado por meio dessas estruturas definidas em espaços conceituais.

De acordo com Gärdenfors (2014a), **conceitos** são estruturas matemáticas capazes de representar o significado das palavras. O tipo mais comum de conceitos envolve as categorias de objetos, embora possam existir também conceitos associados a propriedades, ações, eventos e possivelmente a todas as categorias e combinações de palavras. Em sua teoria, um conceito é definido como uma região convexa de um espaço conceitual.

Para entendermos melhor essa definição, vamos assumir um espaço conceitual como sendo constituído pelo produto cartesiano de um conjunto de **dimensões de qualidade**. Uma dimensão de qualidade pode ser representada por um conjunto ordenado, discreto ou contínuo, de valores que podem ser assumidos por um receptor sensorial referente a alguma qualidade do ambiente que é mensurada por este sensor. Exemplos de dimensões de qualidades podem ser: temperatura, peso, brilho, altura, largura, profundidade, dentre outras. A partir de dimensões de qualidade, podem-se constituir **domínios**, dependendo-se se essas dimensões de qualidade são integrais ou separáveis, conforme a psicologia cognitiva (MELARA, 1992 apud GÄRDENFORS, 2004). Um grupo de dimensões é chamado de integral quando todas as dimensões de qualidade no grupo são igualmente necessárias para caracterizar uma dada qualidade. Cor, por exemplo, é um grupo integral formado pelas dimensões de tonalidade, saturação e brilho⁷. Se assim for, não podemos atribuir brilho a um objeto sem atribuir tonalidade e saturação também. Já um grupo de dimensões que não segue essa definição é separável. Sendo assim, um domínio pode ser definido como um grupo de dimensões integrais que são separáveis de todas as outras. Muitos domínios, tais como a temperatura e peso, consistem em apenas uma dimensão. Outros domínios, como cor e localização, podem exigir múltiplas dimensões. É impossível listar todas as dimensões de qualidade que compõem o espaço conceitual dos seres humanos, uma vez que esse espaço está constantemente se expandindo. Aprender novos conceitos muitas vezes significa expandir o espaço conceitual com novas dimensões de qualidade (SMITH, 1989 apud GÄRDENFORS, 2014a).

Assumindo-se, desta forma, que um espaço conceitual é um espaço n-dimensional constituído pelo produto cartesiano de n dimensões de qualidade, um objeto pode ser visto como um ponto deste espaço conceitual. A partir dessa compreensão, podemos julgar o quanto um objeto é semelhante ou diferente de outros. Dessa maneira, Gärdenfors (2014a) assume que as dimensões de qualidade possuem implicitamente uma métrica, o que nos permite falar sobre distância em um espaço conceitual. Tais distâncias representam o grau de similaridade entre os objetos representados no espaço. Quanto menor a distância, mais semelhante eles são. Gärdenfors (2014a) também enfatiza que a função das dimensões é representar as propriedades de um objeto em diferentes domínios. Uma **propriedade** pode ser entendida como uma predicação que se pode fazer sobre um objeto, com relação a um de seus domínios. Dessa forma, existe uma correlação direta entre uma propriedade e um domínio. Por exemplo, a propriedade “cor” está diretamente relacionada ao domínio da cor, envolvendo por exemplo as dimensões de qualidade de (tonalidade, saturação e brilho). O valor de uma propriedade pode ser visto tanto

⁷ Alternativamente, poderíamos utilizar por exemplo as dimensões de qualidade R, G e B, como sendo as dimensões de qualidade utilizadas em câmeras eletrônicas. Entretanto, Gärdenfors (2004) argumenta que para o olho humano as dimensões de qualidade de tonalidade, saturação e brilho estão mais condizentes com as grandezas físicas mensuradas por cones e bastonetes, as células do olho humano responsáveis pela aquisição dessas dimensões de qualidade

como um ponto, no sub-espço conceitual do domínio referente à propriedade, como um conceito, ou seja, uma região convexa do espaço conceitual formado somente pelos eixos referentes ao domínio da propriedade. Dessa forma, o valor da propriedade “cor”, pode ser, por exemplo, um ponto (t, s, b) correspondente a uma cor específica, ou pode ser “verde”, que corresponde a uma região convexa de pontos do domínio da cor que podem ser categorizados como “verde”.

2.5.1 Categorizações de Propriedades e de Objetos

Na teoria dos espaços conceituais, o domínio é o elemento granular fundamental utilizado para criar o significado dos objetos representados no espaço. Ao identificar as dimensões de qualidade de um domínio, uma região semântica é estabelecida e ela pode então ser utilizada para representar **propriedades** e **objetos**.

Uma **propriedade** consiste de um ponto em um espaço conceitual de um único domínio, que pode possuir uma única ou múltiplas dimensões de qualidade. Um **objeto**, por sua vez, consiste de um ponto em um espaço conceitual que envolve mais de um único domínio. Dessa forma, um objeto poderá ter múltiplas propriedades, dependendo-se do número de domínios de seu espaço conceitual. Tanto propriedades como objetos podem ser categorizados, ou seja, podem ser classificados como pertencentes a diferentes categorias de propriedades ou de objetos.

Tanto **categorias de propriedades** como **categorias de objetos** são conceitos, ou seja, são estruturas semânticas que podem ser utilizadas para atribuição de significado em um espaço conceitual. A diferença entre elas é que uma categoria de propriedade é usada para referenciar um único domínio, enquanto uma categoria de objetos denota a informação contida no correlacionamento de um ou mais domínios. Para entender como uma *propriedade* pode atuar como estrutura semântica, tomemos como base a definição de uma região convexa da álgebra linear.

Dizer que R é uma região convexa significa que para quaisquer dois pontos x e y em R , todos os pontos entre x e y são também pertencentes a R . Mais especificamente, uma **categoria de propriedade** em um espaço conceitual é definida como uma região convexa no domínio associado à propriedade. Se alguns objetos localizados em x e y em relação a uma dimensão de qualidade são exemplos de um conceito, quaisquer objetos localizados entre x e y , em relação à mesma dimensão de qualidade, também serão exemplos deste conceito (GÄRDENFORS, 2004).

As **propriedades**, quando colocadas juntas e correlacionadas de maneira específica, podem ser usadas para representar a **categoria de um objeto**⁸. Segundo

⁸ Por exemplo, considerando cada propriedade como um eixo cartesiano de um espaço n-dimensional, criamos um espaço conceitual n-dimensional, onde regiões convexas desse espaço podem constituir

Gärdenfors (2004), os objetos são identificados como pontos dentro do espaço conceitual e podem ser vistos como um vetor de coordenadas, uma para cada dimensão do espaço. Dessa maneira, os objetos podem ser alocados em regiões específicas dos domínios, onde cada uma representa uma de suas **propriedades**. A categoria deste objeto é, então, indicada por um conjunto de regiões e as suas relações no espaço conceitual.

Existem diferentes estratégias para definir conceitos, ou seja, regiões convexas de um espaço conceitual. Uma das estratégias mais populares é o particionamento de dimensões por meio de protótipos. Vejamos como Gärdenfors apresenta essa estratégia:

Assumamos que as dimensões de qualidade de um espaço conceitual são dadas (por exemplo, as dimensões do espaço da cor, descrito anteriormente) e que nós queremos particioná-lo em um certo número de categorias (por exemplo, categorias de cores). Podemos definir um grupo de protótipos $p_1, \dots, p_n$, um para cada categoria (por exemplo, as cores focais), de tal forma que cada protótipo deve ser o ponto central da categoria que representa. Uma forma de usar esta informação é assumir que, para cada ponto p no espaço, é possível se medir a distância de p para cada um dos protótipos p_i 's, uma vez que o espaço é métrico. Se estipularmos que p pertence à mesma categoria que o protótipo p_i mais próximo, pode-se mostrar que esta regra gerará um particionamento do espaço que consiste de áreas convexas (convexidade aqui definida em termos da medida de distância assumida). Isto é chamado de diagrama de Voronoi [...]. Assim, assumindo que uma métrica é definida no subespaço de categorização, através deste método, um grupo de protótipos gerará um particionamento único do subespaço em regiões convexas. (GÄRDENFORS; WARGLIEN, 2012, p. 8, tradução nossa).

Para ilustrar a explicação feita por Gärdenfors, vejamos a Figura 3.

Neste exemplo, apresentamos um espaço conceitual formado por um único domínio, que é representado pelas dimensões de qualidade X e Y (eixos) e os pontos pretos servem como protótipos para cada categoria. A partir do conjunto completo de pontos, podemos derivar um diagrama de Voronoi, que cria a partição do espaço conceitual, onde cada região compreende uma categoria de propriedade, ou melhor dizendo, uma região semântica que pode ser usada para categorização.

Dessa forma, podemos concluir que a teoria de espaços conceituais é aplicável ao contexto desse trabalho, pois regiões convexas de um espaço conceitual podem ser empregadas na fundamentação de símbolos utilizados para representar categorias de propriedades e de objetos. A pergunta que surge agora é como utilizar os espaços conceituais, de modo que outras categorias gramaticais (além de categorias de propriedades e de objetos) também possam ser modeladas. Como veremos na seção a seguir, Gärdenfors (2014b)

categorias de objetos que possuam tais propriedades. Como um caso especial, podemos considerar que para definir uma categoria de objeto, basta que o objeto possua um mesmo conjunto de propriedades. Ou seja, um domínio inteiro pode ser visto como um conceito. Na verdade esse é o mecanismo típico para gerar categorias de objetos. Entretanto, restrições mais sofisticadas podem ser utilizadas para definir categorias de objetos mais complexas.

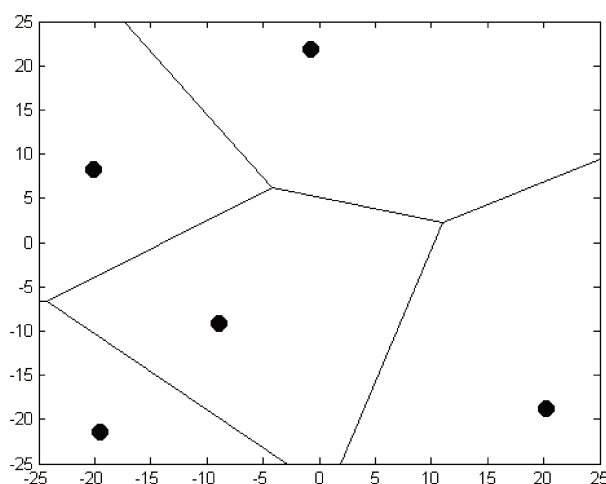


Figura 3 – Diagrama de Voronoi de um espaço conceitual de duas dimensões baseado em 5 protótipos.

também apresenta respostas para esse questionamento, ao propor a Teoria Semântica das Classes de Palavras.

2.5.2 A Teoria Semântica das Classes de Palavras

Um dos conceitos mais importantes em linguística é o de classes de palavras, que pode ser definido como um grupo de palavras que exibem as mesmas propriedades sintáticas, principalmente no que diz respeito às suas flexões e a distribuição em uma sentenças: substantivos, adjetivos, verbos, advérbios, etc. Apesar dessa ser uma definição bastante aceita, [Gärdenfors \(2014b\)](#) afirma não acreditar que a origem das classes de palavras seja realmente sintática, uma vez que as classes que surgem a partir da sintaxe podem ser diferentes, dependendo do idioma. Entretanto, ele acredita que o surgimento de tal divisão em todos os idiomas não seja um mero acaso, mas, ao contrário, tenha origem em padrões universais da cognição humana, e que a correlação com a sintaxe seja um sub-produto e não a origem dessas classes. Em outras palavras, a sintaxe seria um processo emergente em função do surgimento das classes de palavras por meio desses padrões universais.

Para corroborar sua opinião, [Gärdenfors \(2014b\)](#) utilizou o *framework* semântico gerado a partir dos espaços conceituais para mostrar como o significado das principais classes de palavras pode receber fundamentação cognitiva e, assim, serem definidas de maneira genérica (sem definição prévia de um especialista). Vale ressaltar, que não existe um mapeamento direto entre cada classe de palavra e as estruturas do espaço conceitual. O que ocorre, neste caso, é que o espaço conceitual permitirá que se analise e diferencie a região geométrica gerada a partir de conceitos que exibem as mesmas propriedades

sintáticas. Para tal, deve-se deixar clara a diferença entre os conceitos de **categoria de propriedade** e **categoria de objeto**. Dessa forma, Gärdenfors afirma:

A distinção entre categorias de propriedades e categorias de objetos é útil para analisar o papel cognitivo das diferentes classes de palavras. [...], eu proponho que a principal diferença semântica entre adjetivos e substantivos é que adjetivos como *vermelho*, *alto* e *leve* tipicamente referem-se a um único domínio e, assim, representam categorias de propriedades, enquanto substantivos como *cachorro*, *maçã* e *carro* contêm informações sobre vários domínios e, assim, representam categorias de objetos (GÄRDENFORS, 2014b, p. 184, tradução nossa).

Portanto, vejamos a seguir sua proposta para formalização para as principais classes de palavras.

2.5.2.1 Substantivos

Segundo Gärdenfors (2014b), o substantivo é a classe de palavra que se refere às coisas, tais como objetos, pessoas, lugares, substâncias, etc. Mas também existem os substantivos abstratos, que não possuem referentes físicos, como processos, proposições e esquemas que estão fora do tempo e espaço. Além disso, ele ressalta um caso especial dos substantivos que são os nomes. A principal diferença entre eles é que um substantivo expressa um referente, enquanto nomes identificam um único referente. Então a pergunta que surge é: por que todos os objetos, sejam eles físicos ou abstratos, não têm um nome? A resposta que Gärdenfors (2014b) nos apresenta é que os substantivos existem por razões de economia cognitiva, pois seria impossível aprender e lembrar o nome de todos os objetos com os quais interagimos. Assim sendo, os substantivos se referem a grupos de objetos que são suficientemente amplos para a comunicação, de modo que as ambiguidades se tornem raras.

Portanto, a descrição semântica geral que Gärdenfors (2014b) apresenta para os substantivos é de que esses contêm informações de vários domínios e, assim, representam categorias de objetos, as quais são usadas para que não precisemos nomear individualmente cada objeto do mundo. Na teoria semântica das classes de palavras apresentada, os substantivos correspondem à única classe de palavras que depende de mais de um domínio para sua conceitualização.

2.5.2.2 Adjetivos

Substantivos se referem a coisas, verbos dizem algo sobre o que aconteceu a elas. Então por que existem os adjetivos? Gärdenfors (2014b) destaca diferentes formas de sua aplicabilidade na linguagem. A primeira delas é como um especificador do substantivo. Neste caso, o adjetivo ajuda a identificar um referente. Um exemplo apresentado é: “se

você quer que alguém pegue um livro e existem vários livros no contexto, você pode especificar o livro desejado dizendo o ‘livro verde’ ou o ‘livro grande’ ” (GÄRDENFORS, 2014b, p. 186, tradução nossa). Outra forma de usar o adjetivo é como um predicado, como no exemplo: “o fogão está quente”. Neste caso, o adjetivo complementa a sentença caracterizando o substantivo. Ou ainda, em ambos os casos, mas no sentido comparativo, como quando expressamos “a mulher mais alta” ou “Maria é mais inteligente que João” .

Portanto, para Gärdenfors (2014b) a definição semântica de adjetivo é baseada na noção de propriedades, onde a ideia central é que os adjetivos expressam propriedades. Sendo assim, o significado de um adjetivo pode ser representado como uma região convexa em um domínio único.

2.5.2.3 Verbos

O papel semântico dos verbos na linguística é descrito por meio de sua função predicativa. Mas, o que isso quer dizer? Quer dizer que um verbo é o resultado da ligação que se estabelece entre o sujeito e os complementos de uma frase. Entretanto, Gärdenfors (2014b) critica esta definição por ser abstrata demais e, assim sendo, não capturar o papel comunicativo do verbo. Sendo assim, ele propõe uma definição semântica para os verbos a partir da representação cognitiva de **eventos**, pois os verbos são componentes necessários para sua descrição.

Segundo Gärdenfors (2014b), um **evento** é uma estrutura complexa construída a partir de um **agente**, uma **ação**, um **paciente** e um **resultado**. Agentes e pacientes são modelados como objetos com diferentes propriedades, isto é, trata-se de uma categoria de objeto. Além disso, ele assume que os agentes são capazes de atuar, exercendo uma força. Assim, uma ação pode ser modelada como um vetor força ou sequência de vetores força. E por fim, o resultado de um evento pode ser modelado como um vetor de resultado, representando a mudança das propriedades antes e depois do evento. Ele ainda afirma que quando descrevemos um evento pelo menos um vetor força ou resultado e um agente ou paciente devem fazer parte desta descrição.

Portando, para Gärdenfors (2014b), o significado de um verbo surge a partir de uma categoria de evento, gerando uma região convexa formada pelo histórico de mudanças (ou conjunto de vetores) de um domínio único e específico. Sendo assim, podemos exemplificar o verbo empurrar, que se refere a mudanças do domínio força de um evento, ou o verbo mover que se refere a mudanças do domínio da posição, ou ainda o verbo esquentar que se refere a mudanças no domínio da temperatura.

2.5.2.4 Advérbios

Segundo [Gärdenfors \(2014b\)](#), uma das classes de palavras mais desafiadoras de se definir é o advérbio. Pois, da mesma maneira que os adjetivos modificam o substantivo, os advérbios podem modificar todos os demais elementos de uma frase. Sendo assim, podem existir diferentes tipos de advérbios, como advérbios de maneira, advérbios de frequência, advérbios de tempo, advérbios de lugar e outros. Criar uma teoria semântica para cada tipo de advérbio pode ser desafiador demais. Por este motivo, na teoria das classes de palavras, ele irá focar apenas nos advérbios que modificam verbos.

Como visto na seção [2.5.2.3](#), os verbos são modelados a partir de dois tipos de vetores (ou da sequência deles), o vetor força e o vetor resultado. O vetor resultado expressa o que aconteceu no evento e o vetor força como isso aconteceu. Esta relação reflete linguisticamente a relação entre dois tipos de verbos, verbos resultantes e verbos de maneira.

Portanto, para [Gärdenfors \(2014b\)](#) a definição semântica dos advérbios que modificam verbos se referem a alterações no domínio gerado pelas dimensões dos vetores resultado e força, dependendo, assim, de um único domínio.

Como os advérbios dependem de dois tipos de vetores, pode-se diferenciar dois tipos distintos de advérbios. O primeiro deles é o advérbio resultante, que modifica os verbos resultantes. A função deste tipo de advérbio é semelhante àquela dos adjetivos quando modificam o substantivo. Neste caso, esses advérbios funcionam como multiplicadores, amplificando ou diminuindo o valor das dimensões do vetor força, onde o princípio da convexidade pode ser defendido. Por exemplo, se os volumes de vozes $v1$ e $v2$ se referem à falar alto, então qualquer volume entre $v1$ e $v2$ poderá se referir ao advérbio alto. Por outro lado, o segundo tipo de advérbio é o advérbio de maneira, que modifica os verbos de maneira. Esses advérbios expressam a forma de um caminho, descrevendo uma concatenação de mudanças, fornecendo a informação de como este caminho se deu, como nesta sentença: “ela atravessou o parque tortuosamente” (o que significa que atravessou o parque andando em curvas ou em zig-zag).

2.5.2.5 Preposições

A última classe de palavras que [Gärdenfors \(2014b\)](#) apresenta em sua teoria corresponde às preposições. Especificamente, ele as agrupa em duas classes principais: as preposições de localização, que indicam onde algo está, e as preposições de direção, que indicam para onde algo está indo. No caso das preposições de localização, ele afirma que essas modificam o substantivo por especificar a região no domínio espacial, como no exemplo: “Pegue, por favor, a garrafa atrás do pão!”. Já as preposições de direção têm a mesma função de um verbo resultante, isto é, especificar o vetor resultado de um evento,

como no exemplo: “Oscar foi para a biblioteca.”. Em ambas as funções, as preposições são combinadas com um substantivo.

Portanto, para [Gärdenfors \(2014b\)](#) a definição semântica das preposições de localização é representada por grupos de pontos convexos e a das preposições direcionais por grupo de caminhos convexos. Da mesma forma que a definição dos adjetivos e verbos, cada uso de uma preposição também depende de um domínio único. A diferença é que nem todos os usos são baseados no mesmo domínio, o que torna a definição das preposições mais complicada.

2.6 Considerações Finais

Desde que a comunidade científica começou a se preocupar com a maneira como informações percebidas no mundo são abstraídas, registradas na mente de um sistema inteligente e empregadas para gerar conhecimento uma nova área de pesquisa se formou, tratando questões acerca da representação do conhecimento. Neste contexto, diferentes formalismos foram e são propostos por projetistas de sistemas para tratar a tarefa da representação. Entretanto, quando se trata dos mecanismos inerentes à emergência da linguagem, representações simbólicas acabam trazendo consigo o chamado problema da fundamentação do símbolo.

O problema da fundamentação do símbolo é um dos problemas mais antigos da área de inteligência artificial e são muitas as tentativas de propor uma solução para ele. E como o trabalho de Barsalou parece indicar, aquelas soluções que oferecem representações icônicas da realidade, oferecem um caminho alternativo muito promissor para o tratamento deste problema.

Neste sentido, destacamos aqui a teoria do sistema de símbolos perceptuais de [Barsalou \(1999\)](#), principalmente a ideia de simulação mental, que com os símbolos modais, oferece meios para garantir fundamentação simbólica, e a teoria dos espaços conceituais de Gärdenfors, que nos disponibiliza um *framework* semântico capaz de segmentar a realidade em conceitos (ícones) de naturezas diversas, que remetem a diferentes classes de palavras.

De acordo com [Gärdenfors \(2014a\)](#), o uso de espaços conceituais proporciona uma abordagem diferente quando comparado com a IA simbólica. Na IA simbólica, a atribuição de semântica aos símbolos exige uma interpretação externa ao sistema. Na sua proposta, a suposição semântica é de que o significado de uma palavra é definido pelo protótipo da região convexa associada à palavra. Além disso, essa teoria fornece uma estrutura robusta para a aprendizagem de conceitos e sua associação à linguagem, por gerar um espaço dividido em um número finito de regiões semânticas, o que implica num número finito de palavras que podem ser utilizadas para se referir a tais regiões.

Outra característica da teoria dos espaços conceituais diz respeito à sua aplicabilidade, podendo ser utilizada para a definição de uma teoria semântica das classes de palavras. Assim, seu *framework* oferece embasamento para que símbolos referentes a substantivos, adjetivos, verbos, advérbios e proposições possam emergir e ser fundamentados. Essa é uma característica especial para este trabalho de pesquisa, pois de posse de símbolos que remetem a diferentes classes de palavras, símbolos composicionais podem ser empregados.

Sendo assim, ambas as teorias, a do sistema de símbolos perceptuais e a dos espaços conceituais, servirão de embasamento teórico para a modelagem do mecanismo de simulação mental apresentado no capítulo 5. A pergunta que surge agora é em qual contexto ele poderá ser aplicado, a fim de verificar se ele favorece a interoperabilidade semântica. Para isso o utilizaremos nos chamados jogos de linguagem, cuja definição e detalhes adicionais serão apresentados no próximo capítulo (capítulo 3).

3. JOGOS DE LINGUAGEM

Neste capítulo apresentamos o conceito dos jogos de linguagem e alguns experimentos que os implementaram em aplicações computacionais. Embora possam haver vários tipos de jogos, iremos descrever os que mais se destacam na literatura. Dessa forma, em um primeiro momento, falaremos sobre o jogo de adivinhação. Como este jogo agrega em sua formação, os jogos de discriminação e nomeação, optamos por apresentá-los em seções separadas para tornar a leitura mais didática. Por último, falaremos do jogo de ação e algumas considerações a respeito de jogos de linguagem que envolvem símbolos composicionais.

Em todas as seções procuramos fazer uma breve descrição dos jogos e os protocolos que os permeiam. Não entramos em detalhes a respeito dos requisitos e ferramentas computacionais utilizadas nas implementações. Optamos por trabalhar dessa maneira, pois entendemos que cada pesquisador pode ter proposto um ferramental que foi construído a partir de suas concepções filosóficas de pesquisa e isso não interfere na definição dos jogos de linguagem. Entretanto, indicamos no texto as referências bibliográficas para detalhes adicionais.

3.1 Definição

O conceito de “jogos de linguagem” foi proposto pelo filósofo Ludwig Wittgenstein para expressar as condições sobre as quais uma língua se desenvolveria. Para ele os significados de uma palavra não poderiam ser algo pré-estabelecido, onde um nome corresponderia a apenas um objeto e, assim, permaneceria para sempre. Pelo contrário, trataria-se de uma atividade cultural e histórica, que dependeria de um “jogo de linguagem” que estaria sendo jogado o tempo todo em ambientes humanos. Ou seja, em um dado instante de tempo, um nome pode designar um ou mais objetos, e em outro ele pode se referir a objetos completamente diferentes. Tudo dependeria do contexto sob o qual o jogo é desenvolvido.

A ideia de jogos de linguagem rompe com a visão tradicional de que aprender uma língua seria simplesmente dar nomes aos objetos. Esse é apenas um ato secundário de todo o processo, que também envolve a compreensão do ambiente ao seu redor (WITTGENSTEIN, 2009). Na visão de Wittgenstein, uma pessoa aprende também uma determinada forma de vida, contextualizada dentro de uma prática comunicativa interpessoal.

Foi com base nesses princípios que muitos pesquisadores passaram a simular

os jogos de linguagem em meios computacionais para estudar aspectos inerentes dessa habilidade cognitiva que é a linguagem.

Uma das aplicações mais conhecidas na literatura é a proposta por [Steels \(2015\)](#), chamada de *Talking Heads*. Trata-se de uma infraestrutura computacional que emula um sistema cognitivo, como aquele subjacente à linguagem humana, capaz de auto-adaptar durante jogos de linguagens estabelecidos com outros sistemas cognitivos. O objetivo do experimento não é simplesmente a evolução da linguagem, mas estudar e testar a arquitetura cognitiva, composta de mecanismos biologicamente plausíveis, que foi proposta para que tal evolução ocorra.

O experimento consiste de uma população de agentes virtuais incorporados em robôs equipados com câmeras móveis e conectados a um computador para processamento cognitivo e dispositivos para entrada e saída de áudio, como mostra a Figura 4. O ambiente compartilhado consiste em um quadro branco, no qual várias formas geométricas estão coladas e é sobre elas que os agentes irão dialogar. A cada nova configuração do ambiente, um novo jogo será estabelecido, onde categorizações são geradas e nomes são usados para designá-las. Ao final de sucessivos jogos, espera-se que uma linguagem emerja, sem que ela seja estabelecida previamente.



Figura 4 – Configuração do Experimento Talking Head.

Durante cada jogo, um protocolo é estabelecido, configurando o que é chamado de jogo de adivinhação. Neste caso, dois novos agentes são selecionados a partir da população de agentes disponíveis. Um deles fará o papel de falante e o outro de ouvinte, e eles deverão “conversar” sobre a cena que estão visualizando no quadro. O falante escolhe um tópico do quadro branco, conceitua-o e comunica-o verbalmente ao ouvinte. O ouvinte, por sua vez, tem que adivinhar qual tópico foi escolhido. Para isso ele deve

encontrar um significado em seu próprio espaço conceitual e aplicar tal significado na cena para escolher um tópico. Se falhar, ambos os agentes ajustam seus conceitos e nomes utilizados. O processo é repetido e vários agentes agem como falantes e ouvintes, enquanto os elementos no quadro branco estão em constante mudança. A Figura 5, ilustra este protocolo.

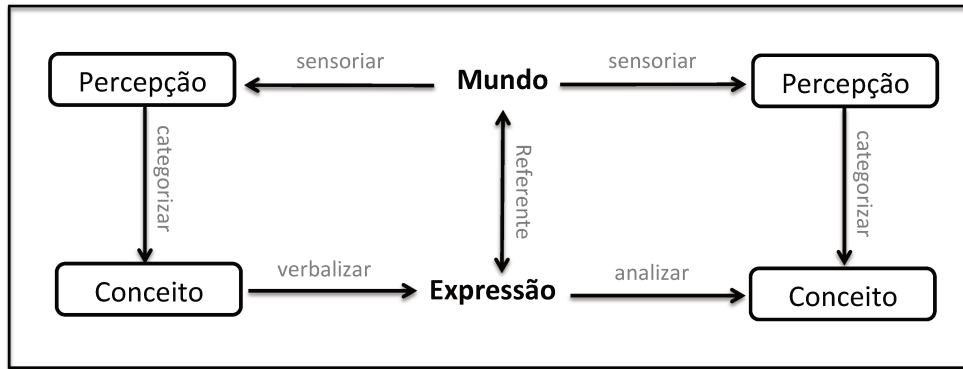


Figura 5 – Protocolo de conversação entre os agentes do experimento *Talking Heads*.

O protocolo integra três módulos distintos, que desempenham os papéis da percepção, da conceitualização e da linguagem. O módulo de percepção tem como objetivo segmentar a imagem percebida a partir de informações vindas dos canais sensoriais. Já o módulo de conceitualização recebe da camada sensorial os sinais discretizados, gerando uma ou mais conceitualizações para estes e as envia para o módulo de linguagem. O módulo de linguagem, por sua vez, deve criar expressões para que os conceitos possam ser verbalizados.

Segundo Steels (2015), como cada agente conceitualiza o mundo conforme sua própria experiência e não tem acesso direto ao conhecimento gerado por outro agente, o experimento *Talking Heads* demonstra como a linguagem auxilia na coordenação dos diferentes conceitos gerados pelos agentes, a fim de se alcançar interoperabilidade semântica entre eles. Dessa forma, se ao final de um jogo emergir um sistema de comunicação bem-sucedido, pode-se concluir que a interoperabilidade semântica se desenvolveu entre os módulos conceituais e de linguagem, estabelecendo um compartilhamento deles entre os agentes. Portanto, o objetivo do experimento é estudar como ocorre a dinâmica entre os diferentes módulos que compõem a arquitetura de um agente, e assim encontrar soluções de engenharia para tal.

Como o jogo de adivinhação envolve um experimento complexo, onde diferentes processos interagem entre si (percepção, conceitualização e linguagem), Steels (2015) optou por dividi-lo. Por esse motivo, é muito comum encontrar na literatura designações de jogos de linguagem como: jogo de discriminação, jogo de nomeação, jogo de ação e outras. Apresentaremos mais detalhes sobre cada um deles nas seções que se seguem.

3.2 Jogo de Discriminação

O jogo de discriminação foi proposto com o objetivo de tratar o problema da categorização, isto é, um mecanismo onde categorias podem ser criadas para que objetos de uma cena sejam distinguidos entre si, o que [Steels \(2015\)](#) chamou de conceitualização. Para ele, essa tarefa consiste em dividir os valores mensurados por sensores específicos em regiões e atribuir uma categoria diferente para cada uma. Por exemplo, um valor de 0.18 da posição horizontal (HPOS) pode ser categorizado como [HPOS 0.0-0.25]. Isso significa que este valor está localizado no primeiro quadrante do canal sensorial e pode ser interpretado como “esquerdo”. Além disso, categorias também podem ser formadas a partir da combinação de valores de diferentes sensores. Neste caso, por exemplo, as categorias “grande” ou “pequeno” podem ser formadas a partir de valores obtidos dos sensores de altura e largura ([BORENSZTAJN; TREHAN, 2006](#)).

No caso específico deste jogo ele deve ser executado por **um único agente**:

- O agente percebe uma cena, que consiste em vários objetos, e escolhe aleatoriamente um objeto. (Neste ponto assume-se que a segmentação foi concluída com sucesso pelo módulo de percepção).
- Em seguida, ele categoriza os objetos na cena usando seu conjunto de categorias (seu espaço conceitual) e recupera uma categoria ou um conjunto de categorias que distingue o objeto escolhido dos outros objetos na cena. Para o agente, este conjunto de categorias distintivas (ou características) constitui o significado do objeto tópico.
- O jogo é bem sucedido se o agente encontrar categorias que servem para identificar o objeto escolhido. Caso contrário, o jogo falhará e o agente é solicitado a produzir mais categorias coerentes e deve tentar novamente.

Vejamos a seguir um exemplo de como pode ocorrer a execução deste protocolo:

1. Um agente **a1** está sensoriando a cena apresentada na Figura 6.
2. **a1** segmenta a cena em 4 objetos: objeto_0; objeto_1; objeto_2; objeto_3;
3. **a1** escolhe o objeto_2 como tópico;
4. **a1** categoriza o objeto tópico como: [VPOS 0.0 - 0.5] (pontuação da característica eleita **0.63**);
5. O jogo de discriminação finalizou com sucesso.

No caso do exemplo apresentado, o jogo finalizou com sucesso, pois o agente foi capaz de encontrar uma característica que diferencia o objeto tópico dos demais. Neste caso, o sensor de posição vertical (VPOS) informou como medida 0.32, o que está dentro

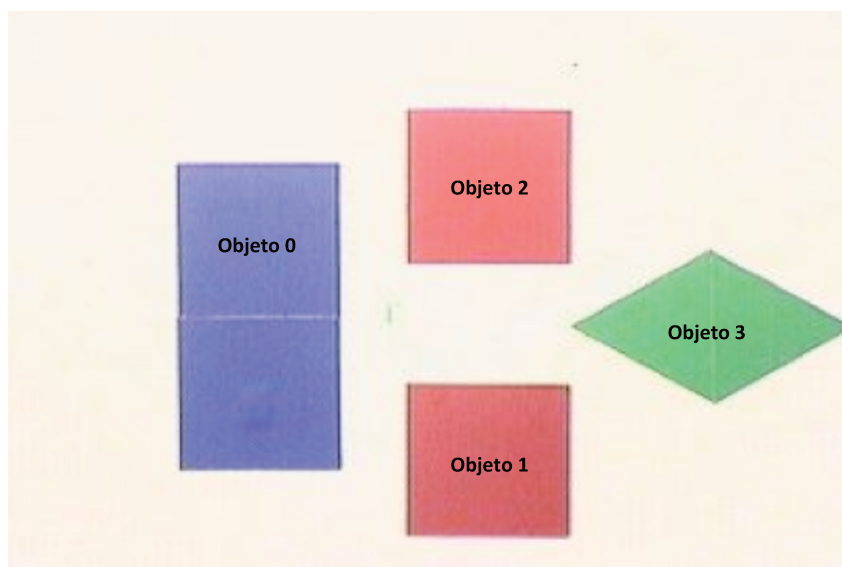


Figura 6 – Cena composta por 4 objetos. Adaptado de Borensztajn e Trehan (2006).

do limite definido pela categoria eleita ([VPOS 0.0 - 0.5]). Sendo assim, pode-se concluir que o tópico é o objeto que está na parte de cima da cena. Se o objeto fosse da cor amarela, por exemplo, “amarelo” também seria uma opção de característica a ser selecionada para distingui-lo e o significado eleito seria aquele que apresentasse a maior pontuação.

3.3 Jogo de Nomeação

Uma vez que os agentes tenham criado os significados/categorias dos objetos do mundo, surge o problema de como nomeá-los para que possam ser verbalizados e ainda, como interpretar um nome verbalizado (isso no contexto do jogo de adivinhação, onde o jogo de discriminação é acoplado ao jogo de nomeação). Ambos os problemas dizem respeito à maneira como os agentes associarão um significado ao seu nome, e é sobre esse contexto que surge o jogo de nomeação. Portanto, o objetivo do jogo de nomeação é demonstrar como um léxico¹ compartilhado pode emergir para que os agentes possam se engajar em interações sociais coordenadas.

Segundo Borensztajn e Trehan (2006), não é trivial desenvolver um léxico compartilhado, pois existem múltiplos agentes e cada um pode inventar um nome diferente para um mesmo significado. Isso poderia resultar em múltiplos nomes para um único significado (sinonímia) ou múltiplos significados para um único nome (ambiguidade). Se não houvesse uma interação dinâmica entre os agentes, e a possibilidade de um agente alterar suas escolhas anteriores de nomes, a partir das informações do ambiente e de outros agentes, cada agente continuaria criando somente seus próprios nomes para expressar

¹ Neste contexto, léxico diz respeito a uma estrutura de dados (também chamada de dicionário léxico) onde associações entre significados e nomes são armazenadas.

algum significado, resultando em sinonímia e ambiguidade generalizadas. Neste caso, os agentes nunca poderiam se entender.

Diferente do jogo de discriminação, o jogo de nomeação é executado através de um duelo entre dois agentes, neste caso um falante e um ouvinte:

1. O falante seleciona um significado do repertório compartilhado de significados, procura um nome para esse significado em seu próprio léxico e transmite esse nome para o ouvinte.
2. O ouvinte interpreta o nome, examinando-o em seu próprio léxico e transmitindo o significado que ele conseguiu.
3. O jogo é bem-sucedido se esse significado corresponder ao escolhido pelo falante, caso contrário, ele falhará.
4. Se falhar, então o falante comunica o significado diretamente.
5. O ouvinte, por sua vez, acrescenta esse novo par nome-significado ao seu léxico.

Em um jogo de adivinhação (Jogo de Discriminação + Jogo de Nomeação), esse protocolo poderia ser descrito da seguinte maneira:

1. Dois agentes, um falante e um ouvinte, são escolhidos aleatoriamente.
2. Uma cena é gerada com um grupo de objetos;
3. Ambos os agentes categorizam cada objeto com base em seus espaços conceituais. (etapa de categorização).
4. O falante, mentalmente, escolhe um objeto referente e recupera seu significado.
5. Se existe um nome para aquela categoria em seu léxico, o falante o recupera. Caso contrário, ele cria um nome para ela. Em seguida ele expressa o nome.
6. O ouvinte escuta o nome e procura por ele em seu léxico. Se houver uma entrada para tal nome em seu dicionário lexical, o ouvinte recupera o significado associado ao nome e tenta apontar o objeto correspondente. Caso contrário, o ouvinte aponta para qualquer objeto para indicar que ele não possui uma categoria associada a esse nome.
7. O falante categoriza o objeto apontado, concordando ou discordando com o ouvinte. Se o objeto indicado possuir o mesmo significado do objeto de referência, o falante indica um sucesso na comunicação e o nome escolhido tem seu peso incrementado. Caso contrário, o falante indica uma falha de comunicação, pune o nome, decrementando seu peso, e indica o objeto de referência correto para o ouvinte.

8. Um novo jogo pode iniciar.

Embora na literatura possam ser encontradas outras versões de jogos de adivinhação, o jogo de nomeação apresentado é suficiente para apresentar a ideia por trás de sua dinâmica. Sendo assim, nas próximas seções apresentaremos mais alguns mecanismos que estão envolvidos nesse tipo de jogo e que são fundamentais para garantir sua convergência.

3.3.1 Sobre o Dicionário Léxico Formado e as Estratégias de Aprendizado

O dicionário léxico é uma estrutura de dados que atua como uma memória associativa, onde pares de significado e nomes são correlacionados. Como ao longo do jogo, o agente ainda está aprendendo, ele deve armazenar significados alternativos para um nome e nomes diferentes para um significado, manuseando respectivamente a ambiguidade (nomes que têm mais de um significado para referenciá-las) e a sinonímia (categorias que têm mais de uma palavra para designá-las).

Para que o jogo evolua e haja uma língua comum entre os jogadores, os nomes associados aos significados recebem uma ponderação a partir de um mecanismo de aprendizado. Um falante só pode transmitir um nome para um significado particular. Dessa forma, ele escolhe aquele com a maior pontuação associada. A pontuação de uma associação é incrementada se for bem sucedida, enquanto a pontuação das associações concorrentes é decrementada, o que [Steels \(2015\)](#) chama de inibição lateral. Se o jogo falhar, as pontuações das associações testadas são decrementadas.

Diferentes estratégias (ou mecanismos de aprendizado) podem ser usadas para ponderar os nomes que estão competindo entre si para descrever um significado. [Wellens \(2012\)](#) elencou várias em sua tese e as principais são citadas a seguir.

- **Jogo de Nomeação Mínimo (do inglês, *Minimal Naming Game* (MNG)):** o agente mantém vários nomes para cada categoria e, no caso de sucesso na comunicação, ambos os agentes removem todos os outros nomes para essa categoria, mantendo apenas o nome do ganhador. Em caso de falha, o agente ouvinte adota o nome proposto pelo falante.
- **Inibição Lateral Básica (do inglês, *Basic Lateral Inhibition* (BLI)):** os agentes mantêm múltiplos nomes para cada categoria, marcando cada nome com um valor real no intervalo $(0,1]$. No caso de falha de comunicação, o falante diminui a pontuação do nome usado com o valor de decremento δ_{dec} e o ouvinte o adota com a pontuação inicial de $s_{inicial}$. No caso de sucesso comunicativo, ambos os agentes aumentam a pontuação do nome usado com o valor de incremento δ_{inc} e diminuem a

pontuação dos outros nomes com o valor de inibição δ_{inib} . Qualquer nome pode ser removido do dicionário léxico caso sua pontuação atinja um valor menor ou igual a 0.

- **Inibição Lateral Interpolada (do inglês, *Interpolated Lateral Inhibition* (ILI)):** é similar à inibição lateral básica. Porém, o incremento (isto é, o reforço) e o decremento (isto é, a inibição) da pontuação (s) é feito a partir de uma interpolação entre 0 ou 1. Para o reforço tem-se $s = s + \delta_{inc}(1 - s)$ e para inibição tem-se $s = s - \delta_{inib}s$.

Uma vez que todas as associações entre significado e nome do dicionário léxico alcancem uma frequência de 100%, um consenso é alcançado. Dessa forma, o léxico da população de agentes não muda mais, a menos que novos agentes ou significados entrem na população.

Os agentes também podem inventar nomes novos, caso eles não tenham um nome para expressar um significado. Eles também podem adicionar nomes novos ao seu dicionário léxico caso ouçam um nome que eles não encontraram antes.

3.3.2 Medindo o Sucesso do Sistema de Comunicação que está se Formando

A métrica mais fácil de ser empregada para medir o desempenho do sistema linguístico que está se formando é chamado de **sucesso comunicativo**. Esta métrica calcula uma proporção média de jogos bem-sucedidos, medindo a capacidade dos ouvintes interpretarem o nome dito por um falante ao indicar o objeto correto associado a ele. Se esse critério for atendido, o sucesso comunicativo é 1, caso contrário, 0.

Além do sucesso comunicativo, utiliza-se também o chamado **alinhamento de nome**. Esta métrica mede o nível de alinhamento da população em relação a um símbolo preferido para cada objeto. Ela depende da competição de nomes, onde a quantidade total de nomes concorrentes por objeto é medida. Assim, Wellens (2012) define:

Dado o conjunto de todos os objetos O , uma população P e uma função de produção $f_{\text{produção}}(o, a)$ que retorna o conjunto de nomes que o agente $a \in P$ prefere para o objeto $o \in O$, tem-se as seguintes fórmulas:

$$\text{competição de nomes} = \langle | \bigcup_{a_j \in P} f_{\text{produção}}(o_i, a_i) | \rangle_{o_i \in O}$$

$$\text{alinhamento de nome} = \frac{1}{\text{competição de nomes}}$$

Uma população está totalmente alinhada quando todos os agentes preferem os mesmos símbolos para cada objeto. Se assim for, o alinhamento do nome é igual a 1.

Outra medida empregada é o **sucesso de alinhamento**. Essa métrica determina a quantidade de agentes que compartilham os mesmos símbolos. Para tanto, ela combina o sucesso comunicativo e o alinhamento de nomes para calcular a média das frequências de todos os pares de significado e símbolo. Assim como o sucesso comunicativo, trata-se de um valor booleano que só é verdadeiro quando o ouvinte conhece o símbolo expresso pelo falante (ou seja, sucesso comunicativo) e é também o seu símbolo preferido para esse objeto. Se todos os agentes preferirem os mesmos pares símbolo-significado para todos os significados, a coerência lexical é de 100%.

3.4 Jogo de Ação

O jogo de ação tem como objetivo evoluir uma língua comum a uma população de agentes, porém voltada para dialetos que permitem entender e executar habilidades motoras. Semelhante aos jogos apresentados nas seções anteriores, a infraestrutura arquitetural subjacente a este tipo de jogo também se molda sobre os três módulos apresentados na Figura 5, desempenhando o papel das habilidades de percepção, conceitualização e linguagem. Entretanto, segundo [Steels et al. \(2012\)](#), esse tipo de jogo envolve desafios adicionais, que são:

1. Para aprender o significado das palavras de ação apresentadas aos agentes, eles devem ser capazes de correlacionar uma ação realizada pelos outros a seus próprios programas de controle motor. Tal habilidade remete a um problema conhecido na literatura da neurociência como o “**problema do espelho**”. O problema do espelho surgiu a partir da descoberta dos chamados neurônios espelhos. Para mais detalhes sobre ele, o leitor pode consultar a seção 13.2 do trabalho de [Steels et al. \(2012\)](#).
2. Falantes e ouvintes devem ser capazes de reconhecer as *affordances*² dos objetos. Para [Steels et al. \(2012\)](#) uma *affordance* surge a partir da combinação da percepção de objetos, das expectativas sobre as possibilidades de ações sobre eles, do conhecimento prévio de suas *affordances* e das pistas dadas pela linguagem.

² O termo *affordance* foi introduzido por [Gibson \(1986\)](#) e envolve o entendimento de quais as ações que podem ser aplicadas a diferentes objetos. Segundo a teoria de Gibson, os objetos nos “convidam” a efetuar ações sobre eles, de acordo com suas possibilidades. Essa característica dos objetos de serem capazes de nos convidar a ações sobre eles seria uma parte importante sobre o próprio modelo do que é um objeto

Assumindo que os agentes podem adquirir diferentes habilidades motoras ao explorarem seu corpo e o mundo, o jogo de ação surge como ferramenta de coordenação das diferentes histórias de interação que os agentes podem formar. [Steels et al. \(2012\)](#) propôs o seguinte protocolo para tratar o primeiro desafio apresentado (o problema do espelho).

Dois agentes são escolhidos aleatoriamente para desempenharem o papel de falante e ouvinte durante uma partida do jogo.

1. O falante escolhe aleatoriamente uma ação de seu repertório de ações.
2. O falante recupera o nome dessa ação em seu dicionário léxico e o transmite para o ouvinte. Se ainda não existir um nome para a ação, o falante poderá inventar um.
3. O ouvinte tenta recuperar a ação solicitada, procurando pelo nome da ação em seu dicionário léxico e evoca o comportamento motor associado a ele. Caso ele ainda não conheça aquele nome, ele escolhe um comportamento qualquer para executar.
4. O falante observa a ação executada pelo ouvinte e verifica se ela combina com a imagem corporal prototípica da ação que ele havia escolhido originalmente. Esse passo requer que o falante tenha uma associação entre a imagem visual de seu corpo com a ação.
5. Caso não haja uma combinação, o falante sinaliza falha. Neste caso, ele ativa seu próprio comportamento motor associado ao nome da ação solicitada, de modo a recuperar o ato comunicativo executado. Dessa forma o ouvinte pode aprender o nome usado pelo falante.
6. Caso contrário, o falante sinaliza sucesso.

Neste protocolo é possível observar que o passo 4 é o mais difícil, pois o ouvinte pode tentar diferentes controles motores e, neste caso, o falante tem que validar todos, até concordar que um deles era o que ele tinha em mente. Esse fato exige que todos os agentes envolvidos nesse tipo de jogo tenham um dicionário léxico sofisticado, que de alguma forma consiga associar o nome atribuído à ação, seu controle motor e seu protótipo visual. Até a convergência dessa estrutura, os seguintes problemas podem ocorrer:

- É possível que o ouvinte já conheça a ação (isto é, seu controle motor e seu protótipo visual), porém ele ainda não conhece o nome atribuído pelo falante;
- O ouvinte pode ter outra ação associada com aquele nome;
- ou ele, de fato, não conhece a ação.

Além do mais, nada impede que esse jogo seja estabelecido entre humanos e agentes robóticos. Neste caso, o humano pode solicitar ao robô que execute uma determinada ação e mostre a ele a execução correta quando este falhar. O contrário também pode ocorrer. Segundo Steels *et al.* (2012), tais formas de aprendizado são muito mais eficazes quando comparadas com formas puramente observacionais e passivas de aprendizado.

Steels *et al.* (2012) ainda propõem outro protocolo de jogo de ação, porém voltado para o problema da *affordance* (o segundo desafio mencionado no início dessa seção). Por envolver a *affordance*, as expressões trocadas durante os jogos possuem uma estrutura composicional (sentença), correlacionando, pelo menos 3 nomes: um para designar o agente, outro para designar o paciente e outro para a *affordance*. Esse fato exige o uso de semântica e sintaxe composicional para que as ações possam ser discriminadas.

Para os trabalhos que exigem essa complexidade, Steels e seu grupo de pesquisa utilizam a ferramenta de Linguagem de Recrutamento Incremental (do inglês *Incremental Recruitment Language* - IRL (ver Spranger (2014))) para o processamento da semântica composicional e a Gramática de Construção Fluída GCF (do inglês *Fluid Construction Grammar* - FCG³ (ver Steels (2017))) para tratar a sintaxe composicional. Ambas as ferramentas fazem parte de um projeto maior e que vem sendo desenvolvido ao longo de anos por este grupo de pesquisa. Juntas elas permitem que os agentes processem sentenças que envolvem ações, bem como participantes e a relação entre eles.

Para validar o protocolo deste jogo de ação estendido, Steels *et al.* (2012) incluem na sua estrutura a utilização de marcadores para as funções dos participantes da sentença. Este tipo de recurso foi usado para estudar como uma gramática de casos, ou seja, uma gramática que marca os papéis dos participantes na linguagem, pode emergir.

Na configuração experimental do jogo, os robôs estão sentados em uma mesa com alguns objetos espalhados. Cada objeto pode ser manipulado de várias maneiras, oferecendo um conjunto de ações possíveis (o que configura sua *affordance*). O falante pede ao ouvinte para realizar uma certa ação e o jogo é um sucesso se o ouvinte executar essa ação. A seguir apresentamos o protocolo para este jogo:

1. O falante observa a cena e escolhe uma ação que ele deseja que o ouvinte execute. Por exemplo, ele quer que o ouvinte empurre o objeto vermelho para o (no sentido de “em direção ao”) objeto azul.
2. Antes de expressar a sentença, o falante constrói um significado para ela usando a ferramenta IRL e, a partir dele, ele constrói o enunciado “empurre o objeto vermelho para o objeto azul” usando sua GCF.

³ Website da ferramenta: <https://www.fcg-net.org/>

3. Ele verifica a existência de alguma ambiguidade. Isto é, ele verifica se a sentença poderia ser interpretada com o significado “empurrar o objeto vermelho para o objeto azul” ou “empurrar o objeto azul para o objeto vermelho”. Em caso positivo, o agente corrige o problema inventando um novo marcador (“-ko”) e o anexa ao paciente da sentença. Sendo assim, o falante diz ao ouvinte: “empurre o objeto vermelho-**ko** para o objeto azul”.
4. O ouvinte analisa o enunciado. Ele ainda não sabe o significado do marcador. Mas, devido aos mecanismos flexíveis de interpretação da FCG e da IRL, ele restringe as possíveis ações para empurrar o bloco vermelho para o azul ou vice-versa. Ele então escolhe uma das ações aleatoriamente. Vamos supor que ele escolhe a correta, então ele empurra o objeto vermelho para o objeto azul.
5. O falante verifica se o ouvinte executou a ação correta. No caso do exemplo apresentado, o falante sinaliza ao ouvinte que ele estava correto.
6. Agora o ouvinte tem certeza de que seu palpite estava correto e ele pode aprender que o marcador “-ko” está marcando o agente da ação empurrar.

Tal protocolo foi executado inúmeras vezes para que uma gramática de casos (gramática que demarca os papéis dos participantes na linguagem) pudesse evoluir. Conforme apresentado por [Steels et al. \(2012\)](#), os resultados foram promissores para o propósito estabelecido⁴.

3.5 Considerações sobre Jogos de Linguagem Composicionais

Os jogos de linguagem apresentados nas seções anteriores são bem estabelecidos na literatura e têm recebido relativo sucesso de aplicação. Embora eles sejam métodos eficientes para associar diferentes símbolos a um mesmo significado, estão longe de representar o que de fato ocorre em uma língua natural. Isto porque o significado dos símbolos são atômicos e estão envolvidos com um único domínio semântico. Além do mais, sentenças formadas por um único símbolo não são capazes de descrever a riqueza de cenários de um contexto do mundo real. Por esses motivos, tais jogos de linguagem são os mais simples que existem ([WELLENS, 2012](#)).

Os símbolos empregados em um sistema linguístico natural, em sua maioria, expressam domínios semânticos distintos e, por esse motivo, são ditos composicionais. [Wellens \(2012\)](#) nos alerta sobre dois aspectos a respeito da natureza desses símbolos. O primeiro deles diz respeito ao fato da composicionalidade ser empregada para descrever

⁴ Mais detalhes sobre as ferramentas e os mecanismos usados, o leitor pode encontrar em [Trijp \(2012\)](#).

objetos a partir dos vários domínios semânticos que os compõem, como por exemplo, “bloco vermelho grande” e para isso não dependem de nenhuma gramática. Já o segundo aspecto, e talvez o mais comum quando se pensa em composicionalidade, diz respeito àquelas sentenças que são formadas por múltiplas palavras e dependem de regras gramaticais para reger sua estrutura, como por exemplo, “Maria empurrou o bloco largo”. Embora em ambos os exemplos a composicionalidade esteja presente, eles possuem suas particularidades e complexidade ao serem empregados em jogos de linguagem. Vejamos um pouco sobre elas nas seções que se seguem.

3.5.1 Símbolos Composicionais que Descrevem Objetos através de seus Domínios Semânticos

Para este tipo de símbolo composicional a complexidade reside no fato de que uma ou múltiplas palavras podem ser usadas para expressar seus domínios semânticos (o chamado problema do significado incerto). Por exemplo, suponha que em um jogo de linguagem o falante queira expressar [**triângulo amarelo grande largo**]. Ele pode fazer isso holisticamente usando uma única palavra, dizendo, por exemplo, “xuzuli”. Assim, essa palavra expressa todos os domínios semânticos mencionados. Mas, ele pode também escolher duas palavras, dizendo “macadu”, cujo significado é [amarelo grande], e “fedizi”, cujo significado é [triângulo largo]. Diante desse cenário, a pergunta que surge é como o ouvinte saberá que “macadu” não significa [amarelo, grande largo] e “fedizi” [triângulo] ou qualquer outra combinação de possibilidades? Como pode ser observado, trata-se de um problema que cresce exponencialmente. Ou seja, quanto mais domínios descritos, haverá mais possibilidades de combinação para expressá-los e interpretá-los. Portanto, ao manipular símbolos composicionais como esses, os falantes do jogo precisam decidir como dividir os domínios semânticos e associá-los às diferentes palavras. Enquanto os ouvintes devem lidar com o problema de encontrar parte do significado expresso por cada palavra.

Esse problema foi estudado por [Wellens \(2012\)](#), que propôs um protocolo de jogo para permitir com que uma população de agentes conseguisse estabelecer um grupo de significados e ao mesmo tempo mapeasse esses significados às suas respectivas palavras. O objetivo do experimento não foi alcançar um sistema lexical totalmente atômico, mas chegar a um que mostre a riqueza dos significados que ele abrange, cobrindo um ou vários domínios semânticos. O protocolo do jogo de nomeação composicional empregado por [Wellens \(2012\)](#) foi o que se segue:

O jogo de adivinhação composicional é estabelecido por dois agentes, um falante e um ouvinte. Ambos percebem um contexto que consiste em n objetos. Cada objeto é composto de k propriedades.

1. O falante escolhe um desses objetos como o tópico e constrói um enunciado, possivelmente composto de várias palavras, que ele acredita chamar a atenção do ouvinte para esse objeto. Nesse caso, ele expressa as propriedades mais discriminativas do tópico em relação aos outros objetos do contexto.
2. O ouvinte analisa o enunciado e tenta interpretar seu significado. Ao final ele aponta para o tópico que ele acredita corresponder ao significado interpretado.
3. Se o ouvinte apontar para o objeto correto, o falante sinaliza sucesso. Se não o falante aponta para o tópico pretendido.
4. Note que o falante apenas aponta para o objeto, nunca para as propriedades que foram expressas.
5. Um novo jogo pode iniciar.

3.5.1.1 Medindo o Sucesso do Sistema de Comunicação que está se Formando

Neste tipo de jogo, o significado das palavras pode ser qualquer subconjunto das propriedades do tópico. O jogo é considerado um sucesso quando o ouvinte aponta para o objeto correto. Dessa forma, para medir o desempenho do jogo, foram empregadas as estratégias descritas no item 3.3.1 e foram extraídos o **sucesso comunicativo** e o **sucesso de alinhamento**, como já apresentados na seção 3.3.2.

Porém, como este jogo envolve composicionalidade, pode ser empregado também o **alinhamento lexical**. Esta métrica mede o nível de alinhamento do uso dos léxicos emergentes. Isto é, a semelhança entre os itens que o falante expressou e aqueles que o ouvinte interpretou. Apesar de sua semelhança com o sucesso comunicativo, ela toma como base a sobreposição do significado associado a cada símbolo da sentença, como mostra a fórmula:

$$\text{alinhamento_lexical}(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

A métrica compara, para cada palavra do enunciado, a semelhança dos significados associados. Trata-se da medida de preferência lexical entre dois agentes ao final de um diálogo. Por exemplo, para o agente A o objeto tópico do diálogo é representado pelo grupo de itens “[amarelo], [pequeno], [triângulo]” e para o agente B também. Isso significa que eles têm a mesma preferência lexical. Enquanto para o agente C, o mesmo léxico é representado pelo grupo de itens “[verde], [grande], [círculo]”, significando que ele tem preferência diferente em relação aos agentes A e B.

No jogo de nomeação apresentado na seção 3.3 o tamanho da população e do contexto influenciavam fortemente o seu desempenho. Porém, no jogo de nomeação

composicional, os resultados mostraram que essas variáveis pouco importavam, sendo, portanto, o número de propriedades que compõe o objeto o fator determinante de seu desempenho. Assim, quanto maior o número de propriedades, maior o grau de incerteza do significado dos enunciados trocados. O experimento apresentado por [Wellens \(2012\)](#) foi satisfatório em mostrar a variedade de significados cobertos por ele.

3.5.2 Símbolos Composicionais Formados por Múltiplas Palavras e que Dependem de Regras Gramaticais

Neste tipo de símbolo composicional a dificuldade reside em outros aspectos da sua formação. O primeiro diz respeito ao significado de cada palavra individualmente e se ela pode ser associada à categoria sintática que compõe a regra gramatical em questão. O segundo aspecto diz respeito ao padrão de ordem da regra gramatical que forma a sentença. Por exemplo, suponha que em um jogo de linguagem o falante queira expressar **[João comeu maçã]**. Ele pode fazer isso usando a regra gramatical formada por [adjunto adverbial + adjunto adnominal + sujeito + adjunto adnominal + verbo + adjunto adnominal + objeto + adjunto adnominal + adjunto adverbial], cuja estrutura requer como único elemento obrigatório o verbo. Dessa forma, ele pode expressar: “repouze tralaleu omotoqui zumaqui”, empregando apenas as categorias sintáticas [sujeito + verbo + objeto + adjunto adnominal].

Para interpretar essa sentença, o ouvinte deve primeiro tentar interpretar cada palavra individualmente, não apenas correlacionando-a com os objetos do contexto, mas também atribuindo um papel sintático a cada uma delas. Depois, o ouvinte deve tentar descobrir qual o padrão de ordem das palavras que foi imposto na sentença. Para cumprir essa atividade, ele deve decidir se a palavra da posição n encaixa na categoria sintática da posição n ou se uma ou mais categorias sintáticas não foram empregadas. Uma vez que as categorias e sua ordem na sentença tenham sido estabelecidas, esta pergunta deve ser fácil de responder. Porém, a medida em que estiverem em evolução, muitos usos inconsistentes podem ser formados e os ouvintes julgarão incorretamente que as categorias foram puladas ou não. Trata-se, portanto, de um problema desafiador.

[Wellens \(2012\)](#) tentou tratar esse problema usando uma gramática muito simplificada, cuja regra é composta de apenas um numeral, adjetivos e um substantivo. Para obter a ordem das palavras da sentença, as estratégias de aprendizado focavam em simplesmente descobrir a categoria sintática a qual o símbolo pertence. Na nossa opinião, esse problema vai além do que ele apresentou. Pois, em primeiro lugar, antes de falarmos da gramática propriamente, deve-se levar em consideração as sentenças e o que elas descrevem no mundo real para evitar o problema da fundamentação do símbolo. E em segundo lugar, a gramática empregada por [Wellens \(2012\)](#) descreve poucos cenários de

um contexto.

O nosso trabalho pretende ir além. Mesmo de posse de uma gramática também simples, temos como intenção descrever cenários mais ricos de contextos simulados e oferecer uma infraestrutura conceitual para que os agentes possam fundamentar as palavras empregadas e correlacioná-las com suas funções sintáticas. Mais do que aprender a função sintática que uma palavra pertence, a nossa intenção é permitir com que os agentes possam alinhar suas regras gramaticais empregadas para formar sentenças composicionais.

Como veremos mais adiante, no capítulo 6, as sentenças criadas por nosso trabalho visam descrever eventos desempenhados pelos agentes sobre pacientes (figuras geométricas) dispostos no ambiente. Assim, as regras previstas para estruturá-las são:

- **Regra 1:** diz respeito à posição do símbolo que remete ao agente. Esse poderá ficar antes ou depois do símbolo que remete a palavra de ação.
- **Regra 2:** diz respeito à posição dos símbolos que descrevem a maneira como a ação ocorreu (isto é, dos modificadores). Há 4 posições possíveis: no início da sentença, antes ou depois do verbo e no final da sentença. A cada posição é atrelado um percentual, que indica a chance dela ser utilizada. Vale ressaltar, que essa posição é definida pelo domínio a que o modificador pertence. Portanto, nem todos os modificadores serão empregados em uma mesma posição.
- **Regra 3:** diz respeito à posição dos símbolos que descrevem as qualidades dos pacientes (isto é, das propriedades). Esses podem ficar antes ou depois do símbolo que descreve o paciente e o agente da sentença. Assim, como na regra 2, esta posição também é definida por domínio ao qual a propriedade pertence.

3.5.2.1 Medindo o Sucesso do Sistema de Comunicação que está se Formando

No contexto do jogo de linguagem composicional, onde há a existência de gramática, além das métricas apresentadas nos outros tipos de jogos (sucesso comunicativo, sucesso de alinhamento, alinhamento de nomes e alinhamento lexical), parece fazer sentido medir também a evolução das regras que estruturam os símbolos ditos. Assim, propomos a seguinte definição:

Dado o conjunto R das regras definidas, uma dupla de agentes $A = a_f, a_o$ (falante e ouvinte) e uma função sintática $f_s(r, a)$ que retorna a probabilidade da regra $r \in R$ para o agente $a \in A$, tem-se a seguinte fórmula:

$$\text{alinhamento sintático} = \langle f_s(r_i, a_f) \times f_s(r_i, a_o) + (1 - f_s(r_i, a_f)) \times (1 - f_s(r_i, a_o)) \rangle_{r_i \in R}$$

Espera-se que de posse dessa fórmula seja possível comparar as regras utilizadas pelos agentes. Uma população de agentes estará sintaticamente alinhada quando todos adotarem as mesmas regras para estruturarem as sentenças trocadas durante o jogo.

3.6 Considerações Finais

Neste capítulo apresentamos os jogos de linguagem, sob a perspectiva do experimento *Talking Heads* proposto por Luc Steels e seus grupo de pesquisa. O intuito de empregar os jogos de linguagem no contexto desse trabalho é utilizá-los como um caso de uso de validação das funcionalidades do mecanismo de simulação mental proposto.

Mas por que empregar jogos de linguagem? Entendemos que quando dois agentes decidem trocar informações entre si não se sabe se eles terão sucesso em tal atividade. Não há garantias de que eles percebem, categorizam, nomeiam e interpretam o ambiente da mesma maneira. Neste contexto os jogos de linguagem tornam-se relevantes, pois atuam como ferramenta de coordenação/mediação entre agentes com diferentes histórias de interação com o mundo. Um tipo de mecanismo que garantirá a interoperabilidade semântica entre os agentes – um objetivo que pretendemos alcançar a partir do nosso mecanismo de simulação proposto.

Os jogos apresentados nessa seção servirão de inspiração para a proposição dos experimentos que iremos desenvolver. O jogo de nomeação será utilizado para a evolução de palavras que denotam classes de palavras como substantivos (comuns e próprios) e adjetivos. Já os jogos de ação englobarão a evolução dos verbos. Os jogos de linguagem composicionais, por sua vez, servirão como mecanismo para englobar em um mesmo protocolo jogos de nomeação e ação, que empregarão sentenças para denotar eventos que descrevem contextos do ambiente simulado e sobre os quais os agentes poderão conversar. A ideia é que ambos os jogos, juntos com o jogo de discriminação, funcionem acoplados em um único jogo de adivinhação.

Na seção 3.3.1 foram apresentadas diferentes estratégias de aprendizado para a convergência de uma linguagem em desenvolvimento. Essas estratégias serviram de inspiração para as que serão propostas nesse trabalho. Elas levam em consideração a dinâmica das interações, de modo que os sinônimos e a ambiguidade possam se dissipar o mais breve possível e um número relativamente pequeno de jogos seja necessário para a evolução da linguagem.

Outro ponto a se destacar é a estrutura de dados utilizada para manter e ponderar cada associação entre as categorias e os nomes criados durante os jogos. A complexidade dessa estrutura de dados será função da classe gramatical da palavra em questão. Por exemplo, nos jogos de nomeação essa estrutura mantém apenas uma as-

sociação entre categorias e nomes. Já no jogo de ação, além dessa relação, também se faz necessário manter o controle motor da ação e, em alguns casos, seu aspecto visual. Sem essa estrutura, que Steels nomeou de dicionário lexical, é impossível que um jogo de linguagem evolua. Tentaremos simplificar essa complexidade, sem deixar de levar em consideração critérios de otimização no momento da busca e inserção dos nomes e categorias.

Por fim, vale ressaltar que diferentes tipos de jogos podem ser estabelecidos, levando em consideração seu escopo e aplicação. Quanto mais complexa a estrutura das palavras/sentenças trocadas entre os interlocutores de um diálogo, mais complexa será o ferramental usado para garantir a evolução do jogo e a emergência da linguagem. Para o caso deste trabalho, propomos um modelo computacional para um processo de representação conceitual e um mecanismo de linguagem. Eles serão apresentados nos capítulos 5 e 6 respectivamente. Entretanto, antes de apresentarmos todo esse *framework* computacional, vejamos no capítulo 4 pontos importantes que também são mencionados por trabalhos correlatos.

4. TRABALHOS CORRELATOS

No capítulo 2 apresentamos as teorias cognitivas que serviram de arcabouço teórico para a proposição do modelo para o mecanismo de simulação mental que será apresentado no capítulo 5. Antes de começarmos a detalhar seu funcionamento, iremos apresentar trabalhos que, assim como este, desenvolveram aplicações computacionais para simular a emergência de símbolos composicionais fundamentados.

Ao iniciar a busca por trabalhos correlatos na literatura, percebemos que o tema de pesquisa abordado vêm sendo estudado em diferentes áreas de atuação. Procuramos concentrar nossos esforços nos trabalhos preocupados com o aspecto semântico dos símbolos composicionais, buscando por modelos gramaticais e *frameworks* computacionais que, de alguma forma, regem os aspectos estruturantes desses símbolos e também se ocupam em estudar os mecanismos empregados para sua geração e interpretação.

Selecionamos 19 trabalhos que julgamos importante referenciar, conforme a categorização da Tabela 1.

Tabela 1 – Divisão dos trabalhos pesquisados por temas abordados.

Contexto de Aplicação do Trabalho		
Área Principal	Com aplicação em	
Frameworks Computacionais Gerais	Arquiteturas Cognitivas	
	Representação Conceitual	
	Sistema Perceptual e Motor	
	Planejamento	
Frameworks Computacionais Aplicados à Linguística	Composicionalidade orientada à sintaxe	
	Composicionalidade orientada à semântica	Modelos Inspirados na Neurociência
		Gramática de Construções
		Gramática de Construções Incorporadas
		Gramática de Construções Fluídas

Primeiramente, faremos uma análise crítica dos *frameworks* de aplicação geral. Mostraremos como este tema é abordado no contexto das arquiteturas cognitivas. Numa

visão mais específica, apresentaremos um mecanismo de representação de conhecimento que pode ser empregado na etapa de conceitualização. A seguir, mostraremos como as informações perceptuais e motoras atuam na fundamentação simbólica ou, ainda, como elas podem ser utilizadas em mecanismos de planejamento. Em seguida, nossa análise se concentrará nos *frameworks* aplicados ao contexto da linguística computacional. Esses trabalhos foram divididos em dois grandes grupos. No primeiro, estão aqueles que tratam a questão da composicionalidade orientada à sintaxe e no segundo grupo, apresentaremos os trabalhos que, apesar de também abordar a questão da composicionalidade, o fazem orientados à semântica. Neste último caso, esses trabalhos foram diferenciados em modelos cuja inspiração vieram da neurociência e aqueles que implementaram computacionalmente uma gramática de construções, além de suas variações: “fluida” e “corporificada”. Para ambos os tipos de trabalhos, faremos uma análise considerando tanto os aspectos negativos (como por exemplo, o não correlacionamento dos símbolos com os elementos que eles representam) quanto os positivos (por exemplo, a fundamentação simbólica feita a partir de interações situadas com o mundo). Para fechamento, destacamos os principais pontos observados dos trabalhos lidos. E por último, para concluir o capítulo, apresentaremos nossas considerações finais.

4.1 Análise Crítica dos Trabalhos Encontrados na Literatura

A partir dos 21 trabalhos selecionados, foi feita uma leitura crítica, destacando principalmente aquilo que cada um se propôs a fazer, bem como seus pontos positivos e negativos. Nos parágrafos que se seguem apresentaremos o resultados dessa análise. Primeiro falaremos dos trabalhos que compõem o grupo dos *frameworks* gerais e, em seguida, dos trabalhos que integram o grupo dos *frameworks* aplicados à linguística.

Como pode ser observado na Tabela 2, 10 trabalhos compõem o primeiro grupo citado. Além da lista de autores, nesta tabela também é possível verificar uma breve descrição de cada um deles. Vejamos mais detalhes nos parágrafos que se seguem.

Ao iniciar a busca por trabalhos correlatos, a preocupação era entender como as arquiteturas cognitivas¹ abordam a questão da emergência de símbolos composicionais fundamentados. Nosso ponto de partida foi um estudo exploratório feito por Samsonovich (2010), que catalogou 26 arquiteturas cognitivas em um *survey*², onde as principais

¹ Segundo Kotseruba e Tsotsos (2018), o tópico “Arquiteturas Cognitivas” é um sub-tópico da Inteligência Artificial, cujo objetivo é criar programas que possam raciocinar sobre problemas de diferentes domínios, desenvolver *insights*, adaptar-se a novas situações e refletir sobre si mesmos. O objetivo final da pesquisa em arquiteturas cognitivas é modelar a mente humana, permitindo construir inteligência artificial semelhante à apresentada por humanos. Por este motivo, as arquiteturas cognitivas fornecem evidências de quais mecanismos conseguem produzir um comportamento inteligente e, assim, contribuem para a ciência cognitiva.

² O estudo de Kotseruba e Tsotsos (2018), mais recente, referencia quase 200 arquiteturas cognitivas.

Tabela 2 – Classificação dos *frameworks* de Aplicação Geral.

Contexto de Aplicação do Trabalho		Autor	Breve descrição
Frameworks Computacionais Gerais	Arquiteturas Cognitivas	Khayi e Franklin (2018)	Simular um sistema de comunicação animal biológico, por meio do qual macacos vervets aprendem o significado de sons de alarme.
	Representação Conceitual	Lieto (2014)	Framework de representação de conceitos para sistemas artificiais inteligentes e arquiteturas cognitivas.
		Lieto et al. (2017)	Para validar as ideias do framework proposto anteriormente, desenvolveram a Dual-PECC (Prototypes and Exemplars-based Conceptual Categorization System), um sistema de representação do conhecimento e raciocínio.
	Sistema Perceptual e Motor	Baillie e Ganascia (2000)	Apresenta uma técnica de aprendizado não supervisionada para extrair informações relevantes (descrições de ações) de sequências de vídeo.
		Siskind (2001)	Apresenta o sistema chamado LEONARD, que produz descrições fundamentais de eventos a partir de seqência de imagens.
	Planejamento	Narayanan (1999)	Trata-se de um modelo computacional para interpretar sentenças causais de domínio político e econômico e gerar inferências.
		Roy (2005)	Framework teórico para guiar um mecanismo de planejamento a partir de interpretações de palavras e atos de fala.
		Frank e Vigliocco (2011)	Modelo conexionista para compreensão de sentenças baseado em simulação mental.
		Polceanu e Buche (2017)	Estudo sobre as abordagens existentes do paradigma de simulação mental, tanto na perspectiva da inteligência artificial quanto na da ciência cognitiva.
		Rocha e Kuc (2019)	Framework de modelagem semântica, capaz de expressar conhecimentos que representam robôs, lugares e objetos.

características dessas arquiteturas foram comparadas. Das 26 arquiteturas mencionadas neste estudo, somente 14 abordavam questões inerentes à linguagem. Entretanto, em sua grande maioria, essas questões se restringiam a aspectos de processamento de linguagem natural, por exemplo, percepção auditiva, análise sintática e semântica de diálogos aplicadas a domínios limitados ou a investigação de aspectos do processamento sintático e semântico de dados textuais. Dentre essas, somente a arquitetura LIDA (Learning Intelligent Distribution Agent) (FRANKLIN *et al.*, 2016) se preocupou com a emergência de símbolos fundamentados, assim como proposto em nosso trabalho. Neste caso, Khayi e Franklin (2018) desenvolveram um experimento para iniciar a inclusão do aprendizado da linguagem na arquitetura LIDA, baseado em experimentos de fundamentação simbólica já relatados anteriormente na literatura (LOULA *et al.*, 2003; LOULA, 2004; LOULA *et al.*, 2010; LOULA *et al.*, 2010). Trata-se de um sistema de comunicação biológico, que simula em um ambiente 2D a maneira como macacos vervets³ aprendem o significado de símbolos vocais. Os agentes filhotes (os aprendizes), controlados por vários módulos cognitivos da LIDA, devem aprender o significado de sinais de alarmes emitidos pelos macacos adultos, associando-os com objetos de seu ambiente externo (predadores), ações de escape e sentimentos de medo. Os resultados apurados foram consistentes com o que se observa nos estudos etológicos, demonstrando que o mecanismo perceptual da arquitetura está apto

³ Os macacos vervets designam uma espécie de macacos encontrados no sul da África.

para aprender conceitos que podem ser usados para fundamentar um símbolo. Entretanto, o experimento, bem como seu antecessor (LOULA, 2004) são capazes de fazer emergir apenas símbolos isolados, não considerando questões como composicionalidade simbólica, sintaxe e gramática.

Corroborando com os trabalhos de Loula (2004) e Khayi e Franklin (2018), Lieto (2014) também demonstrou a importância do sistema perceptual para a emergência de símbolos composicionais fundamentados. Neste caso, ele propôs um *framework* de representação de conceitos para sistemas artificiais inteligentes e arquiteturas cognitivas. O *framework* propõe uma abordagem heterogênea para a representação de conceitos, parcialmente inspirado na teoria de protótipos proposta por Eleanor Rosch. Ao invés de possuir um único meio de representação, segundo Lieto conceitos podem ser representados tanto por meio de protótipos, como exemplos ou teorias. A principal contribuição do trabalho de Lieto (2014) é a apresentação de uma proposta heterogênea para o debate do problema da representação do conhecimento, fazendo conexões inexploradas entre diferentes teorias. Originalmente, suas ideias eram apenas teóricas, deixando para o futuro uma aplicação prática aplicada ao contexto do aprendizado léxico. Alguns anos depois, Lieto *et al.* (2017), desenvolveram a arquitetura cognitiva Dual-PECCS (Prototypes and Exemplars-based Conceptual Categorization System), uma extensão da arquitetura cognitiva Dual incluindo mecanismos heterogêneos de representação de conceitos, cujo objetivo principal é extrapolar a capacidade dos sistemas artificiais nas tarefas de categorização. O modelo de representação da arquitetura Dual-PECCS foi integrado também nas arquiteturas ACT-R e CLARION, e empregado em tarefas de descrição linguística e prototipação. Para representar o conhecimento usado nas tarefas, foram usados como ferramental espaços conceituais e ontologias, conferindo ao sistema a capacidade de manipular conceitos de diferentes naturezas. Apesar de Lieto *et al.* (2017) abordarem o problema da representação conceitual, eles não informam como esses conceitos podem ser aprendidos. Isso de certo modo pode ser uma limitação para o contexto da linguagem, pois o significado dos símbolos pode depender de quem está projetando o sistema. Além do mais, os conceitos abordados evocam palavras que designam itens lexicais (isto é, símbolos isolados) e não se sabe ao certo como aplicar o sistema para representar conceitos que designam, por exemplo, transformações temporais (ações, modificações no tempo e etc).

Baillie e Ganascia (2000) demonstraram como informações perceptuais e motoras podem ser extraídas e representadas mentalmente para modelar a semântica de símbolos que designam palavras de ação. Eles empregaram descritores quantitativos para descrever ações a partir de informações extraídas de sequências de vídeo. Apesar da técnica tomar como base o reconhecimento do perfil do evento (uma técnica bastante empregada para o reconhecimento de ação), a proposta deles não foi capaz de criar um descritor genérico para categorizar um conjunto amplo de ações ou ações elementares, como eles haviam proposto.

De maneira semelhante, [Siskind \(2001\)](#), desenvolveu o sistema LEONARD capaz de ler sequências de imagens para produzir descrições de eventos. O diferencial desse sistema é o emprego da dinâmica de forças entre objetos, fornecendo, assim, uma representação conceitual para ação subjacente ao evento descrito. O próprio [Gärdenfors \(2014a\)](#) reconhece o modelo de forças como o mais adequado para representar ações. Além disso, o sistema também é capaz de detectar hierarquias de eventos. Porém, ele não se mostrou genérico o suficiente para explorar outras informações do evento que não fossem a ação e os objetos participantes.

Uma outra abordagem interessante foi a apresentada por [Roy \(2005\)](#). Ele propôs um *framework* onde palavras e atos de fala são interpretados para gerar informações úteis para o mecanismo de planejamento. Esse trabalho, além de abordar os símbolos sob a perspectiva semiótica de Peirce, emprega simulações mentais, geradas a partir de uma rede de esquemas e signos, para fundamentar símbolos. Porém, atua numa abordagem unidirecional. Isto é, preocupa-se apenas com a interpretação de símbolos. Outra limitação, diz respeito ao fato dele embutir no sistema os conceitos usados para fomentar os esquemas empregados na simulação mental, sem explicar como eles foram adquiridos. Além do mais, a proposta apresentada no trabalho ainda não foi implementada na prática.

Também com o intuito de gerar informações úteis, [Narayanan \(1999\)](#) propôs um modelo computacional para interpretar sentenças causais de domínio político e econômico para gerar inferências. A versatilidade deste modelo decorre do fato dele usar um tipo de rede de Petri modificada, capaz de modelar dinamicamente informações aspectuais e temporais do domínio encenado. Tal recurso atua como um mecanismo de simulação mental robusto. Isso explica o porquê desse modelo ser agregado à gramática de construções corporificada, como veremos adiante. Entretanto, assim como o trabalho de [Roy e Reiter \(2005\)](#), este trabalho tem como limitação o fato de não ser bidirecional. Isto é, trata apenas a questão da interpretação dos símbolos. E embora também empregue o conceito de simulação mental para gerar as inferências situadas e corporificadas, não se sabe de onde o conhecimento usado para gerar as projeções metafóricas foi obtido. Isso nos leva a acreditar que o modelo também depende de seu projetista para funcionar.

É importante mencionar também o trabalho de [Frank e Vigliocco \(2011\)](#) que propõem um modelo connexionista para formalizar a teoria do sistema de símbolos perceptuais de [Barsalou \(1999\)](#). Diferente de outras abordagens, o modelo é capaz de criar representações analógicas não somente para gerar inferências, mas também para resolver ambiguidade. Porém, assim como os outros trabalhos, que empregam simulação mental para fundamentação simbólica, não se preocupa com a questão de como os conceitos usados para gerar as simulações são aprendidos. Além do mais, é unidirecional (atua somente na perspectiva da interpretação simbólica).

Ainda pensando na habilidade cognitiva de planejamento, porém aplicado na

área de robótica, Polceanu e Buche (2017) desenvolveram um vasto estudo sobre as abordagens existentes do paradigma de simulação mental, tanto na perspectiva da inteligência artificial quanto na da ciência cognitiva. Eles conseguiram elencar as principais lacunas na literatura existente sobre simulação mental e forneceram algumas sugestões para seu desenvolvimento futuro. De modo geral, eles observaram que o mecanismo de simulação mental comumente vem sendo empregado para que agentes possam encontrar soluções para cenários não triviais em ambientes virtuais ou físicos, sem aplicações em linguagem, propriamente.

Corroborando com esse achado, recentemente Rocha e Kuc (2019) apresentaram um *framework* de modelagem semântica, capaz de expressar conhecimento intrínseco, a fim de melhor representar robôs, lugares e objetos. Os dados gerados eram usados para gerar automaticamente uma simulação mental completa, permitindo que os robôs pudessem simular a si mesmos e outros agentes modelados no ambiente. As simulações geradas permitiram que robôs realizassem o aprendizado e planejamento autônomos, sem a necessidade de modelos personalizados.

Esses foram, portanto, os trabalhos que compuseram o grupo dos *frameworks* de aplicação geral. Vejamos a seguir os trabalhos que compõem os *frameworks* aplicados à linguística. É neste grupo que se concentra o maior volume de material consultado, somando 11 no total. Assim como foi feito para o grupo anterior, na tabela 3 também pode-se conferir seus autores e suas descrições.

Tabela 3 – Classificação dos *frameworks* Aplicados à Linguística.

Temas Abordados			Autor	Breve descrição
Área Principal	Com aplicação em			
Frameworks Computacionais Aplicados à Linguística	Composicionalidade orientada à sintaxe		Vogt (2005)	Modelo para estudar a evolução e indução de estruturas composicionais na linguagem em uma população de agentes robóticos simulados, atuando no contexto dos jogos de linguagem.
			Sierra-Santibáñez (2014)	Também utiliza jogos de linguagem para simular evolução de um sistema linguístico para descrição de construções lógicas.
	Modelos Inspirados na Neurociência	Glenberg e Gallese (2012)	Adaptaram dois modelos de controle motor (MOSAIC e HMOSAICO) para propor o modelo ABL (action-based language), capaz de aprender (substantivos, verbos e sentenças composicionais) compreender e produzir (gestos) linguagem.	
		Gramática de Construções	Dominey et al. (2006)	Modelo para processamento de construções gramaticais.
			Madden et al. (2010)	Modelo de Compreensão Híbrida que fundamenta construções e discursos em seus desdobramentos temporais provenientes de simulações situadas, controladas pela gramática.
	Lallee et al. (2010)		Propôs uma extensão do trabalho de madden et al (2010), incluindo no framework a representação teleológica (raciocínio orientado a objetivos).	
	Composicionalidade orientada à semântica	Gramática de Construções Incorporadas	Bergen e Chang (2013)	Gramática de Construções Incorporadas (ECG - do inglês Embodied Construction Grammar) é um formalismo representacional de gramática que emprega modelos incorporados para sua aprendizagem e uso.
		Gramática de Construções Fluidas	Steels (2017)	A Gramática de Construções Fluidas (FCG - do inglês Fluid Construction Grammar) trata-se de uma plataforma computacional para desenvolver gramáticas de construções e usá-las em experimentos computacionais para estudar entendimento, produção e emergência de linguagem.
			Spranger e Steels (2012)	Aplicação computacional usada para explicar a origem das estruturas composicionais, funções gramaticais e classes de palavras que descrevem relações espaciais.
			Steels et al. (2012)	Aplicação de jogos de linguagem , mais especificamente jogos de ação, para que agentes robóticos possam formular e interpretar comandos que especificam ações que o corpo pode realizar (por exemplo, levantar) ou ações que envolvam outros objetos de uma cena (por exemplo, pegar bloco vermelho).
			Trijp (2012)	Experimento para demonstrar a origem de uma gramática de casos. O trabalho demonstra experimentalmente a efetividade da gramática de casos para descrever eventos.

Nesse estudo exploratório encontramos uma série de trabalhos envolvidos com os mecanismos responsáveis pela formação e interpretação de sentenças linguísticas. Esses trabalhos tinham como principal característica a composicionalidade (propriedade das

expressões complexas cujo sentido é determinado pelo de seus constituintes e pelas regras usadas para combiná-los). Entretanto, quando abordada a partir de mecanismos computacionais, essa composicionalidade pode ser compreendida a partir de dois vieses diferentes: o sintático e o semântico.

Os trabalhos que tratam a composicionalidade a partir de um viés sintático presumem que ela pode ser aprendida a partir da análise de padrões estruturais extraídos de vários exemplos de sentenças armazenadas em um banco de dados. Neste caso, encontramos o trabalho proposto por [Vogt \(2005\)](#) que, seguindo a mesma linha dos trabalhos de [Kirby \(2000\)](#), propôs um modelo para estudar a evolução e a indução de estruturas composicionais da linguagem em uma comunidade de agentes simulados, atuando no contexto dos jogos de linguagem. Com este trabalho [Vogt \(2005\)](#) conseguiu demonstrar a evolução de uma gramática comum entre dois agentes, capaz de gerar não apenas estruturas composicionais, como também estruturas holísticas. Seu mecanismo de indução é versátil, e funciona tanto para gerar as sentenças, quanto para interpretá-las. Entretanto, como a gramática empregada é arbitrária, não se sabe se ela funcionaria para manusear símbolos demarcados por papéis sintáticos, semelhantes àqueles de uma língua natural.

Da mesma forma, [Sierra-Santibáñez \(2014\)](#) empregou jogos de linguagem para a evolução de um sistema gramatical visando a formação de sentenças lógicas, incorporando o mecanismo a uma população de agentes com 10 indivíduos. Os agentes empregam fórmulas lógicas para conceitualizar as sentenças geradas, o que nos leva a concluir que o trabalho atua sob a perspectiva simbólica da IA clássica. Portanto, ele acaba implicando no problema da fundamentação simbólica, onde os símbolos não tem relação com as entidades do mundo que designam.

A análise feita sobre esses dois trabalhos nos levou a compreender que este segmento tem como principal objetivo entender como as regras que regem a formação das sentenças podem emergir ou serem aprendidas. Porém, como o foco da nossa pesquisa não se restringe a esse aspecto apenas, optamos por não aprofundar tanto nessa temática. Dessa maneira, nos concentramos nos trabalhos que tratam a composicionalidade a partir de um viés semântico. Isto é, aqueles que acreditam que a composicionalidade surge porque a semântica empregada para estruturar uma sentença também é composicional. Assim, basta entender a semântica dessas estruturas complexas que as regras que regem sua formação serão compreendidas também.

Iniciamos nossos estudos por um trabalho que tomou como inspiração modelos da neurociência. Neste caso, [Glenberg e Gallese \(2012\)](#) adaptaram dois modelos de controle motor, denominados MOSAIC e HMOSAICO, para propor o modelo ABL (action-based language) capaz de compreender e produzir substantivos, verbos e sentenças composicionais. Assim como a maioria dos trabalhos correlatos que analisamos, este parte da premissa de que o significado da linguagem surge a partir de interações situadas

com o mundo. Entretanto, o seu diferencial decorre do fato de usar informações advindas do sistema motor, enquanto a maioria dos trabalhos exploram o sistema perceptual apenas. Para tal, emula o conceito de neurônios-espelho. Outra vantagem deste trabalho é o fato de manipular em uma mesma aplicação tanto símbolos computacionais, quanto isolados. Além disso, é bidirecional, isto é, realiza tanto a interpretação de símbolos, quanto sua geração. Porém, o vocabulário linguístico é limitado aos comportamentos que o módulo motor do sistema é capaz de executar.

Já no contexto da linguística propriamente, analisamos trabalhos que fazem uso de gramáticas, mais especificamente a gramática de construções, para processamento de linguagem natural. Antes de apresentar tais trabalhos, cabe explicar o que é a Gramática de Construções.

A gramática de construções é um modelo da linguística cognitiva, proposto a partir da década de 80, por Fillmore e colaboradores para designar que o léxico e a sintaxe de uma língua não constituem módulos separados, mas formam um sequência de construções que parte de elementos específicos (como a palavra “cadeira” ou uma expressão idiomática como “bater as botas”) até padrões mais abstratos (como um adjetivo). Como apresenta [Ferrari \(2011\)](#), esta gramática propõe que expressões linguísticas, desde as mais simples até as mais complexas, constituem signos simbólicos de acordo com a visão Saussureana, correspondentes ao par forma (significante) e significado. Esse pareamento é descrito por meio de esquemas que capturam características compartilhadas por várias sentenças específicas. Por exemplo, as sentenças “Ela dançou samba” e “Ele perdeu a cabeça” são ambas instâncias da construção transitiva [SN1 V SN2], onde os sintagmas nominais SN1 e SN2 são complementos do verbo e exercem, respectivamente, as funções de sujeito e objeto direto. Trata-se, portanto, de uma visão que explica a regularidade da gramática com base em esquemas gerais e não a partir de regras algorítmicas para a manipulação de símbolos, como proposto pela gramática gerativa.

Com o intuito de explorar suas vantagens computacionalmente, [Dominey et al. \(2006\)](#) propuseram um modelo neurolinguístico para interpretar construções gramaticais. A novidade apresentada por este modelo é uma descrição computacional de como os humanos fazem o mapeamento entre forma e significado ao tentarem interpretar uma sentença linguística. Entretanto, como o modelo sócnico manipulado pela gramática de construção é diádico, não há relação do símbolo com as coisas que ele designa no mundo. Portanto, acaba implicando no problema da fundamentação do símbolo.

Para tratar esse problema, [Madden et al. \(2010\)](#) fizeram uma adaptação ao modelo neurolinguístico de [Dominey et al. \(2006\)](#), fundamentando as construções a partir de desdobramentos temporais provenientes das simulações mentais situadas, controladas pela gramática. Este, então, passou a ser designado como um modelo de compreensão híbrido, permitindo tratar a cooperação e a negociação entre robôs e humanos, por meio de

troca de sentenças. Como se trata de um modelo unidirecional (preocupa-se apenas com a compreensão das sentenças) e emprega simulação mental para fundamentação simbólica, mais uma vez temos um caso de onde os conceitos usados para gerar as simulações não são aprendidos, dependendo de uma pré-configuração feita por seu projetista.

Em seguida, [Lallee et al. \(2010\)](#) propuseram uma extensão do trabalho de [Madden et al. \(2010\)](#), incluindo no modelo a representação teleológica (raciocínio orientado a objetivos). A representação teleológica é uma componente crucial para o modelo, pois permite representar subcomponentes das ações, o que inclui relações entre o estado inicial e final de sua execução, permitindo dessa forma, a interpretação de construções gramaticais mais complexas (conectivos causais, tipo porque, se-então). Entretanto, as desvantagens apresentadas pela versão anterior do trabalho ainda não foram tratadas nessa versão.

Desde a proposição da ideia de uma gramática de construções, especializações foram propostas, onde é possível, atualmente, citar pelo menos 5 variações deste modelo linguístico: gramática de construções baseada em signos (do inglês, *Sign-Based Construction Grammar*) ([BOAS; SAG, 2012](#)); gramática cognitiva (do inglês, *Cognitive Grammar*) ([LANGACKER, 2009](#)), gramática de construções radicais (do inglês, *Radical Construction Grammar*) ([CROFT, 2001](#)), gramática de construções corporificada (do inglês, *Embodied Construction Grammar*) ([BERGEN; CHANG, 2013](#)) e gramática de construções fluidas (do inglês, *Fluid Construction Grammar*) ([STEELS, 2017](#)). Dentre essas, as que mais encontramos aplicações em simulações computacionais para manipulação de sentenças composicionais fundamentadas na literatura foram a gramática de construções corporificada (a partir de agora citada no texto como GCC) de [Bergen e Chang \(2013\)](#) e a gramática de construções fluidas (a partir de agora citada no texto como GCF) de [Steels \(2017\)](#).

Diferentemente de outros trabalhos dessa área, tanto a GCC quanto a GCF têm como objetivo principal fornecer implementações computacionais das estruturas e processos linguísticos da gramática de construções. Para isso, se faz necessário propor toda uma infraestrutura para que o conhecimento linguístico possa ser representado, empregado e, no caso de interesse dessa tese, como os significados devem ser conceituados ou interpretados pelos usuários, a partir de sua própria percepção sensório-motora do mundo.

Embora fundamentados sob os mesmos pilares conceituais, a GCC e a GCF surgiram a partir de diferentes perspectivas e com diferentes objetivos. A GCC foi projetada para apoiar a modelagem cognitiva de aprendizagem e uso da linguagem natural por meio de simulações corporificadas. Já a GCF é um formalismo para representar gramáticas e suportar experimentos computacionais de jogos de linguagem que abordam a evolução da comunicação em populações de agentes robóticos.

Bergen e Chang (2013) propuseram a GCC para implementar um ferramental computacional capaz de gerar um conjunto de inferências de maneira dinâmica e que tivesse conteúdo semântico. O modelo possui como fonte de conhecimento um rico repositório de construções que expressa informações linguísticas por meio de um esquema de forma e significado. Esses esquemas são empregados em dois processos: a análise e a simulação. Dados uma sentença e um contexto, o processo de análise determina o grupo de construções que devem ser responsáveis por eles. Após realizar uma avaliação é gerada uma especificação semântica, que determina quais são os esquemas conceituais evocados através das construções eleitas e como elas são correlacionadas. A partir da especificação, o processo de simulação irá explorar suas representações subjacentes, para simular eventos específicos, ações, objetos, relações e estados (aqui tem-se o modelo de simulação incorporado proposto por Narayanan (1999)). O ponto forte dessa gramática é sua abrangência, pois os esquemas que configuram as construções cobrem várias particularidades da uma língua natural, gerando inferências fundamentadas sobre experiências perceptivas e motoras. Entretanto, uma crítica que podemos fazer diz respeito à dependência que a simulação tem do repositório das construções, que neste caso, parece que deve ser configurado previamente. Além do mais, trata-se de uma plataforma poderosa, porém aplicada apenas para a interpretação da linguagem (é unidirecional).

Não obstante essas limitações, temos que destacar a importância dessa gramática na área da linguística. Além de servir de plataforma para diferentes tipos de aplicações, ela vem influenciando outros trabalhos com objetivos similares. Podemos citar como exemplo o trabalho que vem sendo desenvolvido por um grupo de pesquisa da Universidade Federal de Juiz de Fora. Da mesma forma que o analisador da GCC, Laviola *et al.* (2017) desenvolveram um sistema que visa fornecer análises computacionais, cognitivamente plausíveis, do significado de sentenças. Ele faz interpretações de sentenças do português brasileiro a partir de frames e construções definidas no banco de dados do projeto FrameNet Brasil. Para tal, atua diferenciando construções que possuem a mesma sintaxe, porém significado diferente.

Steels (2017), por sua vez, idealizou a GCF com o intuito de se tornar um formalismo computacional para representar a gramática de construções de maneira rápida e fácil. Esse fato tem chamado a atenção de linguistas do mundo inteiro. Além disso, a GCF possui uma série de características que a tornam adequada para experimentos que exploram o surgimento de sentenças composicionais: (i) aplicação de regras bidirecionais para formação e interpretação de sentenças; (ii) fluidez na aplicação dessas regras (isto é, regras podem ser criadas e modificadas constantemente), o que gera uma competição entre elas em uma população de agentes. Entretanto, para que as regras possam gerar sentenças fundamentadas, a gramática depende de um sistema a parte, denominado IRL (do inglês, *Incremental Recruitment Language*).

O IRL é um sistema para planejamento de significado de estruturas simbólicas composicionais, que une a percepção e o processamento da linguagem, a partir de representações geradas por um sistema que está conectado ao mundo real. Ao receber uma sentença para ser interpretada ou ao se criar uma para expressar algo, o IRL executa uma série de operações (que devem ser configuradas a priori) para gerar uma rede semântica. Sobre ela, inicia-se um processo de busca heurística para encontrar estruturas que produzem categorias altamente discriminativas. Essas categorias, então, podem ser expressas pela regras da GCF ou usadas para interpretar uma sentença que foi estruturada a partir delas.

Apesar da GCF apresentar um alto poder de representação, semelhante ao das diversas nuances de uma língua natural, a complexidade de seu *framework* deixa a desejar. Pois, além de aprender a manusear o ferramental, deve-se configurar as regras que irão compor as construções e também as operações do sistema IRL para que as categorias semânticas possam emergir durante a produção e interpretação linguística.

De todo modo, alguns membros do grupo de pesquisa de Steels desenvolveram uma série de experimentos para o contexto abordado neste capítulo e que se faz necessário apresentar aqui.

[Spranger e Steels \(2012\)](#), por exemplo, propuseram uma aplicação de jogos de linguagem para explicar como estruturas composicionais, funções gramaticais e classes de palavras que descrevem relações espaciais podem emergir. Neste experimento, a GCF foi fundamental para permitir que o ouvinte das sentenças empregadas pudesse escolher as estratégias de conceitualização corretas nas redes criadas pelo IRL.

Outra aplicação interessante foi a desenvolvida por [Steels et al. \(2012\)](#) que empregaram jogos de ação para que agentes robóticos pudessem formular e interpretar comandos que especificam ações que o corpo pode realizar (por exemplo, levantar) ou ações que envolvam outros objetos de uma cena (por exemplo, pegar bloco vermelho). Apesar de apresentar resultados não tão inovadores, eles trouxeram o conceito de jogos de ação, juntamente com ações de natureza diversa (monovalentes, bivalentes e trivalentes) e os marcadores de papéis sintáticos.

Por fim, [Trijp \(2012\)](#) desenvolveu um experimento para demonstrar a origem de uma gramática de casos⁴. O trabalho demonstra experimentalmente as vantagens cognitivas da gramática de casos para descrever eventos. Porém, a aplicação não é genérica. Para funcionar em outros contextos precisa estar acoplada à GCF.

Aqui encerramos a análise crítica dos 19 trabalhos selecionados para serem apresentados nesse capítulo. Como forma de finalizar essa seção e também deixar registrado todo o mapeamento que foi feito, deixamos a Tabela 4 que contém um sumário

⁴ A gramática de casos é um sistema linguístico que analisa a estrutura das sentenças estudando os papéis semânticos que são requisitados pelo verbo.

completo de tudo que foi discutido anteriormente. Ela pode ser verificada no anexo [A](#) deste trabalho.

4.2 Considerações Finais

Neste capítulo procuramos reunir os principais pontos dos trabalhos que estão envolvidos com processos que atuam no mecanismo de emergência de símbolos composicionais fundamentados (sentenças). Como pode-se perceber, os trabalhos selecionados apresentam uma pluralidade de temas e áreas. Isso aconteceu porque tínhamos o intuito de responder as seguintes perguntas:

1. Como a emergência simbólica vem sendo tratada na literatura?
2. Esses símbolos são de qual natureza, isolados, composicionais, abstratos?
3. Numa perspectiva semiótica, qual modelo sógnico é usado (diádico, triádico)?
4. Os conhecimentos perceptual e motor têm sido usados para fundamentar esses símbolos?
5. Em caso positivo, esses conhecimentos têm sido usados para aprender categorias que podem auxiliar no processo de categorização das coisas a que esses símbolos se referem?
6. Os conceitos de alguma forma auxiliam na fundamentação dos símbolos? Eles podem evocar classes de palavras diferentes?

A emergência simbólica diz respeito aos processos de formação e interpretação de símbolos, mais especificamente, à maneira como um símbolo é criado, empregado por uma comunidade de falantes e seu significado torna-se comum a eles. Trabalhos desenvolvidos pelo grupo de Luc Steels (ou até mesmo de Vogt), por exemplo, seguem essa perspectiva. Porém, a maioria dos trabalhos nem sempre trata esse aspecto da linguagem. Normalmente, utilizam a linguagem para extração de conhecimento para aplicações diversas (raciocínio e planejamento, simulação de processos neuronais, troca de informações entre robôs e humanos, compreensão de línguas naturais e outras mais). Essa diversidade de aplicabilidade faz com que esses nem sempre se ocupem em manter a plausibilidade biológica dos meios empregados para que o símbolo ganhe significado.

No que diz respeito à fundamentação de símbolos isolados, a literatura têm alcançado relativo sucesso, desenvolvendo modelos que atuam de forma semelhante aos seres vivos quando fazem uso dessa habilidade (ver, por exemplo, [Hutchins e Hazlehurst \(1995\)](#), [Jung e Zelinsky \(2000\)](#), [Oliphant e Batali \(1997\)](#), [Oudeyer \(1999\)](#), [Steels \(1999\)](#),

Roy (2002), Vogt e Coumans (2002), Loula (2004)). Entretanto, o mesmo sucesso não é observado nas referências que tentam explorar os aspectos gramaticais da linguagem ao empregarem símbolos composicionais, especialmente no caso de sentenças que contém um verbo. O que observamos nesse caso, é que diferentes frentes de pesquisas vêm sendo conduzidas para tentar eliminar essa limitação.

Na linguística cognitiva, por exemplo, observamos que alguns trabalhos vêm empregando esforços em tal situação. Entretanto, alguns deles ainda não conseguiram correlacionar as sentenças com os referente do mundo. Os símbolos manipulados são diádicos, trazendo consigo o problema da fundamentação, como no caso dos trabalhos de Sierra-Santibáñez (2014) e Dominey *et al.* (2006). Por esse motivo, percebemos uma tendência dessa área de pesquisa em empregar informações perceptuais e motoras através de simulações mentais (conforme o modelo de Barsalou (1999)) para a fundamentação de símbolos (Glenberg e Gallese (2012), Madden *et al.* (2010), Lallee *et al.* (2010), Bergen e Chang (2013)). Os resultados têm sido positivos, tanto que frameworks de aplicação gerais também estão fazendo uso dessa tendência (Narayanan (1999), Roy (2005), Frank e Vigliocco (2011)). Destacamos aqui o trabalho de Roy e Reiter (2005), que é o único a citar signos triádicos, de acordo com a perspectiva de Peirce. De todo modo, duas críticas podem ser feitas sobre esses trabalhos. A primeira decorre do fato dessas aplicações se preocuparem apenas com a interpretação de sentenças, deixando de lado a sua formação. Isso é um problema, pois as sentenças manipuladas dependem de uma gramática a parte para que possam ser estruturadas. O outro problema diz respeito às simulações geradas por trabalhos da linguística. Barsalou (1999) é categórico ao afirmar que as informações perceptuais e motoras extraídas das simulações mentais são evocadas a partir de experiências passadas de um indivíduo. Como essas aplicações não tomam esse cuidado, elas dependem do projetista do sistema para configurar as informações que alimentam essas simulações. No caso da CGI, proposta por Bergen e Chang (2013), mesmo as simulações sendo configuradas a partir das informações extraídas das construções, vale ressaltar que elas compõem uma base que foi escrita à parte.

Outra forma de se explorar informações perceptuais e motoras foi aquela apresentada nos trabalhos de Baillie e Ganascia (2000) e Siskind (2001). Neste caso, elas foram utilizadas para representar conceitualmente ações e eventos. Trabalhos como esses têm uma relevância para nosso tema de pesquisa, pois apresentam técnicas inovadoras para tratar a conceitualização das transformações temporais dos objetos do mundo, o que normalmente é o mais complexo de se fazer. Mas diferente da nossa abordagem, eles exploram somente a ação, deixando de fora outros elementos envolvidos nessa dinâmica e que também poderiam ser expressos na linguagem. E por falar em conceitos, temos que destacar aqui trabalhos como o de Lieto *et al.* (2017) que apresentam novas perspectivas para o problema da representação conceitual no contexto das arquiteturas cognitivas e como os conceitos podem ser empregados para fundamentar símbolos linguísticos. Outra

inovação deste trabalho foi empregar espaços conceituais e ontologias como ferramentas de representação, atuando juntas, em um mesmo problema. Trabalhos como esse tem grande relevância, por propor mecanismos que são fundamentais para que arquiteturas cognitivas possam trabalhar a emergência de símbolos fundamentados. Talvez seja por essa carência que atualmente, esses *frameworks* computacionais, em sua grande maioria, vêm empregando a linguagem da mesma maneira como sistemas tradicionais da área de processamento natural de linguagem.

Voltando à questão da emergência simbólica, conseguimos apresentar aqui alguns trabalhos que, de fato, atuaram nessa frente de pesquisa. No caso do trabalho de [Vogt \(2005\)](#), foram pesquisadas como regras que regem a estruturação dos símbolos composicionais podem emergir numa população de falantes e podem se tornar comum entre eles. Isto é, [Vogt \(2005\)](#) pesquisou a emergência de regras sintáticas para sentenças composicionais simples. Talvez os resultados mais interessantes, e que tenham mais semelhança com a nossa proposta, sejam os trabalhos do grupo de Luc Steels. Todos eles tratam a composicionalidade a partir de uma perspectiva semântica. E por esse motivo propõem uma gramática de construções fluidas, que ancorada no sistema IRL, é capaz de evoluir redes semânticas empregadas na conceitualização das sentenças estruturadas por essa gramática. Dessa forma, a gramática cuida dos aspectos sintáticos das sentenças e o IRL de seus aspectos semânticos. Na nossa opinião, este grupo foi o que conseguiu apresentar os resultados mais expressivos para a área de emergência de linguagem, além de, por meio da GCF, cobrir uma série de particularidades que são encontradas nas línguas naturais. Isso se deve aos diversos anos de pesquisas que este grupo vêm devotando ao tema. Entretanto, durante os momentos iniciais de nossa pesquisa, fizemos uma tentativa de empregar os *frameworks* propostos por eles e nos pareceu complexo demais aprender a manusear dois sistemas, um especificamente em linguagem LISP (no caso do IRL), depois configurá-los e aplicá-los no problema abordado. Foi então que nasceu a proposta desta tese, cujo intuito é também tratar a emergência de símbolos composicionais, a partir de um único *framework* e ancorado sobre uma gramática limitada, mas que oferecesse ao sistema a capacidade de manipular sentenças mais simples, mas principalmente denotadas pelo verbo. Acreditamos que essas sentenças, mesmo sendo mais simples, conseguirão conferir ao sistema certo poder comunicativo.

Por fim, vale ressaltar ainda os seguintes pontos. Não conseguimos encontrar nenhum trabalho que abordasse a fundamentação simbólica de símbolos abstratos, como aqueles que designam sentimentos. Isso ainda parece ser um problema de pesquisa em aberto. Rolf Zwaan ([ZWAAN, 2016](#)) vem apresentando afirmações contundentes de que a solução para este problema só será possível a partir do emprego de abordagens mistas (a partir das abordagens simbólica, conexionista, situada e corporificada e até espaços conceituais) para a fundamentação simbólica. Essa é uma abordagem interessante a se pesquisar. Outra observação é em relação à natureza dos símbolos evocados pelas

categorias usadas na fundamentação. Não encontramos nenhum trabalho que empregasse categorias conceituais para fundamentar símbolos que remetessem a diferentes classes de palavras, além de substantivos e adjetivos. Essa é uma das questões abordadas por este trabalho de pesquisa.

5. MECANISMO COMPUTACIONAL PARA SIMULAÇÃO MENTAL

Considerando o estudo feito no capítulo 2 sobre o problema da fundamentação do símbolo e sua relação com as teorias do sistema de símbolos perceptuais e espaços conceituais, apresentamos neste capítulo uma proposta de implementação computacional para um mecanismo de simulação mental, que permitirá a criação e interpretação de símbolos gramaticais.

A originalidade da nossa proposta decorre do fato da implementação empregar espaços conceituais como framework semântico subjacente ao processo de fundamentação simbólica. Como visto no capítulo anterior, todos os trabalhos que fizeram uso do mecanismo de simulação mental no contexto da linguagem não formalizaram seu funcionamento através de um modelo, e muito menos associado à ideia de espaços conceituais.

Embasar nosso modelo em espaços conceituais confere a ele a capacidade de representar a realidade através de diferentes tipos de categorias ontológicas, o que na nomenclatura de Peirce nos permite afirmar que criamos um mecanismo de representação icônica da realidade. De modo mais específico, entendemos que essa representação pode ser considerada uma metáfora, um tipo de ícone segundo a teoria Peirceana.

Diferente de Platão, Kant e Peirce, que propuseram teorias de categorias universais, não temos a pretensão de que a camada de representação gerada por nossa proposta seja universal, ao analisar todos os aspectos da realidade. Pelo contrário, entendemos que o conjunto de categorias ontológicas proposto serve ao propósito de fazer com que uma mente artificial seja capaz de usá-las, abstraindo uma grande parte dos fenômenos que nós, humanos, conseguimos compreender do mundo real. Mesmo havendo coisas que uma máquina não vai ser capaz de entender a partir dessas categorias, permitir que máquinas consigam entender pelo menos parte das coisas que entendemos já será um grande feito. Este foi um dos objetivos que tínhamos em mente ao propor tal conjunto de categorias e que não encontramos em nenhum trabalho revisado.

Portanto, a contribuição da nossa pesquisa é a proposição de um modelo computacional para o mecanismo de simulação mental, empregando um conjunto de classes ontológicas para descrever diferentes fenômenos do mundo real, de tal forma que uma criatura artificial possa se utilizar das mesmas para criar um modelo mental do mundo ao seu redor. Esse modelo servirá de fundamentação para os símbolos linguísticos criados em uma linguagem emergente entre agentes inteligentes. Dada a riqueza dos fenômenos descrito por eles, símbolos linguísticos que remetem a diferentes classes de palavras

são criados e interpretados, conferindo às criaturas também a capacidade de empregar linguagem gramatical.

Como se tratam de duas atuações distintas por parte do modelo - uma que diz respeito ao emprego das categorias ontológicas para criar modelos mentais e outra para traduzi-los por meio da linguagem gramatical - optamos por apresentar neste capítulo os aspectos arquiteturais envolvidos com as estruturas de representação e o processo perceptual. Já a apresentação do mecanismo de simulação mental, na perspectiva da linguagem, bem como seu mecanismo de evolução, serão apresentados em detalhes no Capítulo 6.

5.1 A Modelagem Computacional para o Mecanismo de Simulação Mental Proposto

A Figura 7 apresenta o *framework* computacional que foi desenvolvido para representar o mecanismo de simulação mental proposto. Trata-se de um conjunto de classes de software que foram utilizadas na construção dos agentes capazes de simulação mental. Essas classes podem ser organizadas em 5 pacotes: *Raw*, *Conceptual*, *Ontology*, *Perception* e *Language*.

A escolha por tal organização computacional mantém a plausibilidade do *framework* com o referencial teórico utilizado e, principalmente, com o pressuposto de nossa hipótese de pesquisa. Assim, o pacote *Raw* será responsável pela representação bruta dos objetos sensoreados, o pacote *Conceptual* pela gestão do espaço conceitual, bem como a definição dos domínios e dimensões de qualidades e o pacote *Ontology* pela definição das categorias ontológicas empregadas na categorização da realidade. Dessa maneira, as informações brutas sensoreadas podem ser traduzidas nas categorias ontológicas previstas (propriedade, categoria de objeto, objeto conceitual, *affordance*, modificador, evento e episódio) sem depender previamente do conhecimento de um especialista, pois são aprendidas a partir das regiões semânticas formadas pelo espaço conceitual. Como forma de exercitar esse correlacionamento, o modelo também contém o pacote *Language*. Ele provê o arcabouço computacional necessário para traduzir as categorias ontológicas detectadas em símbolos linguísticos e, no sentido inverso, realizar a tradução desses símbolos nas categorias ontológicas correspondente. Mais adiante, forneceremos mais detalhes sobre as classes de cada pacote e como elas interagem entre si para compor o funcionamento do mecanismo de simulação mental.

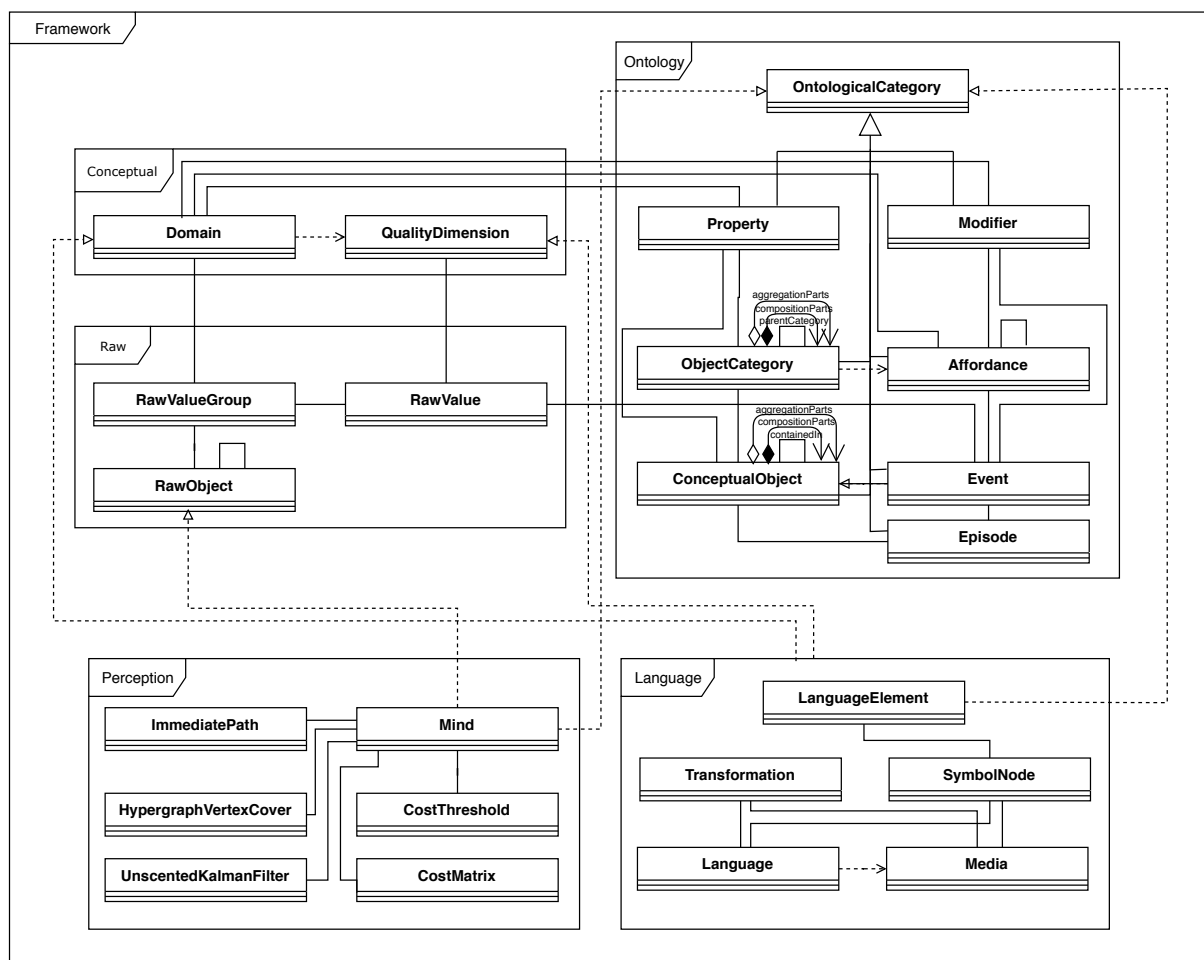


Figura 7 – Diagrama de classes simplificado do *framework* de simulação mental proposto.

5.1.1 Sobre os Pacotes de *Software*

5.1.1.1 Pacote *Raw*

O pacote *Raw* contém todas as classes responsáveis por captar as informações que são recebidas pelos sensores e formatá-las na mente do sistema, de modo a formar uma representação bruta do objeto ou evento que será categorizado. Como mostra a Figura 7, ele é composto de três classes principais: **RawValue**, **RawValueGroup** e **RawObject**.

A classe **RawValue** é utilizada para definir um valor bruto para uma dimensão de qualidade. Por exemplo, o número real “25.0” pode representar o valor bruto atribuído para a dimensão de qualidade “graus celsius” que compõe o domínio da temperatura.

A classe **RawValueGroup** é utilizada para agrupar valores brutos de um domínio. Por exemplo, “255.0, 0.0, 0.0” é um valor bruto do domínio da cor, cujas dimensões de qualidade expressam “vermelho”, “verde” e “azul”. Assim teria-se como representação de um **RawValueGroup**: cor: [255 (R), 0 (G), 0 (B)]

A classe **RawObject**, por sua vez, representa a descrição bruta de um objeto

por meio de seus **RawValueGroups**. Essas informações brutas são obtidas diretamente a partir dos sensores. Por exemplo, “cor: [255 (R), 0 (G), 0 (B)]; tamanho: [5 (X), 5 (Y), 5 (Z)]; forma: [6 (vértices)]” podem ser um **RawObject** resultante do sensoreamento de um cubo vermelho do ambiente.

5.1.1.2 Pacote *Conceptual*

O pacote *Conceptual* é composto pelas classes **Domain** e **QualityDimension**, mantendo referência direta com os dois principais elementos formadores do espaço conceitual, na definição de Gärdenfors.

A classe **QualityDimension** tem por objetivo representar uma dimensão de valores sensorizados, correspondendo a uma dimensão de qualidade.

Já a classe **Domain** é formada por uma lista de dimensões de qualidade, que é utilizada para compor um domínio. Por exemplo, as dimensões “R, G e B” podem ser as dimensões de qualidade integrais que formam o domínio “cor”. Já o domínio “tamanho”, por sua vez, pode ser formado pelas dimensões de qualidade “altura, largura e profundidade”. Ela também possui uma lista de propriedades e modificadores que correspondem às regiões semânticas identificadas durante o processo de descobrimento de categorias.

5.1.1.3 Pacote *Ontology*

O pacote *Ontology* é formado pelas classes **Property**, **ObjectCategory**, **ConceptualObject**, **Affordance**, **Modifier**, **Event** e **Episode**. Cada classe permite que diferentes tipos de categorias ontológicas possam ser aprendidas a partir de regiões semânticas de um espaço conceitual e servirão para fundamentar o significado de diferentes classes gramaticais usadas na linguagem.

A classe **Property** utiliza as regiões convexas de um domínio para representar as propriedades. Dessa forma, por exemplo, a propriedade *vermelho* pode se referir a qualquer cor (domínio formado pelas dimensões “R, G, B”) cuja dimensão de qualidade predominante seja a dimensão R. As propriedades descobertas pelo mecanismo de simulação estabelecerão no pacote de linguagem o papel semântico de adjetivos.

A classe **ObjectCategory** foi criada para definir o conceito de uma categoria de objetos. Seguindo Gärdenfors (2014b), categorias de objetos contêm informações de vários domínios e são usadas para que não precisemos nomear individualmente cada objeto. Sendo assim, uma categoria de objeto estabelecerá no pacote de linguagem o papel semântico dos substantivos comuns.

A classe **ConceptualObject** tem uma função similar à da classe **ObjectCategory**. Porém, ao invés de fundamentar os substantivos comuns, ela estabelecerá no

pacote de linguagem o papel semântico dos substantivos próprios. Segundo [Gärdenfors \(2014b\)](#), este conceito deve existir para atribuir um nome a uma entidade que mereça destaque. Ambas as categorias devem existir, pois nem todo conceito de objeto receberá um nome específico por motivos de economia de memória. Ela é formada por um conjunto de domínios e suas respectivas propriedades, além de outros dois atributos do tipo **ObjectCategory** que seriam suas partes composicionais e agregadoras.

Para fundamentar a representação semântica dos verbos foi criada a classe **Affordance**. O termo “*affordance*” foi estabelecido por [Gibson \(1986\)](#) para designar possibilidades de ação disponíveis em objetos para que um indivíduo (independente de sua habilidade para perceber estas possibilidades) possa interagir com eles. Na modelagem proposta, a classe **Affordance** possui como atributo o domínio onde ocorre a mudança associada à *affordance*. Além disso, existem: duas categorias de objetos, que representam, respectivamente, o agente que pode aplicar esta *affordance* e o paciente que pode sofrê-la, a lista dos modificadores que ela admite (no parágrafo seguinte explicaremos com mais detalhes o que é e para que serve um modificador).

A classe **Modifier** foi criada para definir o conceito do elemento que modifica o sujeito, objeto, ou verbo de uma sentença. Sendo assim, ela estabelecerá no pacote de linguagem o papel semântico dos advérbios que modificam os verbos resultantes. No processo de percepção de eventos, esta classe será usada para comparar o valor bruto das propriedades do objeto que está sofrendo a ação, no início e no final de um evento, gerando um vetor resultante.

A classe **Event** foi criada para representar nosso modelo semântico de estruturas mais complexas, tais como uma sentença, e será instanciada toda vez que uma possibilidade de ação/*affordance* for aplicada. Nesse modelo, uma sentença envolve um ou mais objetos e um único verbo, que modifica temporalmente alguma propriedade dos objetos envolvidos. Seguindo [Gärdenfors \(2014b\)](#), um evento pode ser formado a partir de um agente, uma ação, um paciente e um resultado, tal como “João moveu o bloco amarelo rapidamente para a esquerda”. Os agentes e pacientes são vistos como objetos com diferentes propriedades. Sendo assim, devemos interpretá-los como instâncias de categorias de objetos (ou objetos conceituais, dependendo do contexto). Já a ação é a atuação do agente, exercendo uma força (ou uma sequência de vetores força), o que nós interpretamos como sendo uma *affordance*. E, por último, o resultado de um evento seria o vetor de mudança, representando a alteração das propriedades do agente e/ou paciente antes e depois da aplicação da *affordance*. Neste caso, interpretamos o resultado como um advérbio, representado semanticamente pela classe **Modifier**.

Por fim, a classe **Episode**, integra um conjunto de **Events** que possuem uma coesão temática comum, representando nosso modelo semântico para múltiplas sentenças concatenadas, que descrevem uma cena tematicamente coesa, como em um fragmento

de texto mais longo, envolvendo mais de uma sentença. Uma instância de um **Episode** descreve um conjunto ordenado (ou paralelo, ou em superposição temporal parcial) de eventos centrados em um ou mais objetos que compõem o tema de uma cena. Um exemplo de episódio seria: “João pegou a maçã. Maria cortava cebolas. João comeu a maçã. João cuspiu as sementes Maria chorou”.

5.1.1.4 Pacote *Perception*

No pacote *perception* estão as classes responsáveis pelo sensoreamento, rastreamento e categorização dos objetos. Ele é composto pelas classes: **Mind**, **ImmediatePath**, **UnscentedKalmanFilter**, **CostThreshold**, **CostMatrix** e **HypergraphVertexCover**.

A classe **Mind** é um modelo da mente de um agente. Ela agrega todas as funcionalidades do *framework* e utiliza classes de todos os pacotes, inclusive as categorias ontológicas, para realizar o rastreamento de objetos, suas categorizações e o aprendizado linguístico.

A classe **ImmediatePath** é empregada para modelar caminhos (*paths*) de rastreamento dos objetos na memória imediata do agente.

A classe **UnscentedKalmanFilter** implementa a versão “*unscented*” do filtro de *Kalman* apresentada por [Julier e Uhlmann \(1997\)](#). Ela é empregada para remover ruídos dos valores brutos sensoreados e também para construir representações brutas estimadas para continuação de caminhos criados anteriormente na memória imediata.

A classe **CostThreshold** é uma classe auxiliar que foi criada apenas para armazenar os menores valores de um conjunto de números e indicar qual limiar separa esses valores dos demais. Os limiares calculados por esta classe são utilizados para eliminar caminhos ou passos que não estejam dentre os de menor custo.

Por fim, a classe **HypergraphVertexCover** implementa um algoritmo usado para encontrar a menor quantidade de “hiper-arestas” que cubram todos os vértices de um “hiper-grafo”. Um sistema a utiliza para escolher o menor conjunto de propriedades discriminativas para expressar simbolicamente um objeto conceitual.

5.1.1.5 Pacote *Language*

O pacote *Language* é composto pelas classes que irão fornecer todo o ferramental necessário para que uma proto-linguagem possa emergir e evoluir entre seus falantes. Nelas estão contidos os artefatos computacionais que permitiram como que uma categoria ontológica possa ser expressa linguisticamente, por meio de símbolos composicionais. Ela também contém os artefatos que realizam a interpretação dos símbolos.

A classe **SymbolNode** define a construção de cada símbolo manipulado pelo mecanismo.

A classe **Media** descreve um tipo de mídia que pode ser utilizado em um sistema de comunicação. Neste caso, a mídia pode ser do tipo texto, áudio, sinal ou outras. Vale ressaltar que mídias distintas de um mesmo idioma serão tratados como sistemas de comunicação distintos.

A classe **Transformation** é usada para definir as transformações que podem ser aplicadas nos símbolos da linguagem que está em desenvolvimento. Entende-se como transformações as conjugações que um símbolo pode sofrer, tais como a transformação de tempo (presente, passado ou futuro), de número (singular ou plural) e de propriedade (aumentativo ou diminutivo). Atualmente, no modelo proposto, as transformações possíveis de serem empregadas são as de tempo (neste caso, passado e futuro) e a de número (somente a do tipo plural).

A classe **Language** é a classe mais importante do pacote, pois irá definir as configurações do mecanismo para que a proto-linguagem possa se desenvolver. Para tanto, ela agrega em sua estrutura um atributo de cada classe citada anteriormente e os métodos de evolução da linguagem, que permitirão com que símbolos possam ser criados e interpretados, bem como também as regras para reforçar o aprendizado gramatical e sintático.

Por fim, a classe **LanguageElement** faz a ligação entre os símbolos e as categorias ontológicas que eles designam. Dessa maneira, torna-se possível a associação entre um símbolo e seu significado, na forma de uma categoria ontológica.

Detalhes adicionais a respeito dos aspectos computacionais de cada pacote mencionado podem ser consultados no Anexo B desta tese.

5.1.2 Interações dos Pacotes de Software Durante o Ciclo Cognitivo da Simulação Mental

Como visto, cada pacote de software que compõe o *framework* computacional apresentado tem uma função específica no ciclo cognitivo do mecanismo de simulação mental. O correlacionamento entre eles ocorre por meio de processos intermediários que em conjunto conferem ao mecanismo de simulação dois modos de atuação distinto - o modo percepção e o modo linguagem. Como mostra a Figura 8, esses processos realizam as seguintes tarefas (na figura eles são destacados com a cor amarela):

1. Modelar informações sensoreadas;
2. Descobrir categorias ontológicas;

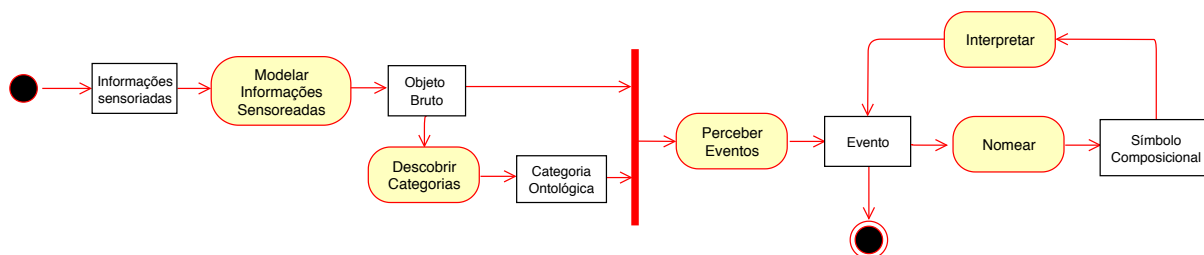


Figura 8 – Processos que compõem o ciclo do mecanismo de simulação mental proposto.

3. Perceber eventos;
4. Nomear eventos;
5. Interpretar símbolos linguísticos.

De modo geral, o correlacionamento entre os processos inicia-se a partir das informações captadas pelos sensores. Essas são repassadas para o processo de modelagem de informações que, a partir das classes do pacote *Raw*, vai formatá-las, gerando um objeto bruto. O objeto bruto vai ser empregado pelo processo de descobrir categorias ontológicas que, a partir das classes do pacote *Conceptual*, segmenta o espaço conceitual e emprega suas regiões semânticas para reificar os diferentes tipos de categorias ontológicas previsto pelo pacote *Ontology*. Os objetos brutos sensoreados, juntamente com sua categoria ontológica identificada, são manipulados pelo processo de percepção que, a partir de algoritmo de rastreamento, verifica se uma *affordance* foi aplicada. Caso positivo cria-se um evento para registrar como ela ocorreu. É esse processo que define a atuação do mecanismo de simulação mental no modo percepção. O evento é empregado pelo processo de nomeação para criar um símbolo composicional para expressá-lo. Este símbolo pode ser empregado pelo processo de interpretação para restaurar as categorias que fazem sua fundamentação simbólica e que permitem restaurar o conjunto de sensações que as originaram. Esse processo, marca o fim da atuação do mecanismo de simulação proposto. Vale ressaltar que os processos de nomeação e interpretação definem a atuação do mecanismo de simulação mental no modo linguagem.

Embora a atuação desses 5 processos tenha sido apresentada de maneira linear, na realidade não é bem assim o que acontece. Essa estratégia foi adotada para deixar a explicação mais didática. Como a implementação do *framework* computacional emprega o conceito de *threads* não há garantia de que os processos ocorram sequencialmente, um após o outro.

Entretanto, é importante destacar que há um certo grau de dependência entre eles. Por exemplo, o processo de perceber eventos só poderá acontecer uma vez que diferentes tipos de categorias ontológicas estejam disponíveis para categorização. Assim, ele depende do processo de criar categorias. E ambos os processos (criar categorias e

perceber eventos) dependem dos objetos brutos que são gerados na modelagem de informações sensoreadas. O mesmo relacionamento pode ser observado entre os processos de nomeação e interpretação. A nomeação só poderá gerar símbolos a partir de eventos já criados e a interpretação dependerá desses símbolos para restaurar os eventos que fazem sua fundamentação.

A seguir vejamos mais detalhes da atuação de cada processo mencionado.

5.1.2.1 Sobre a Modelagem de Informações Sensoreadas

A modelagem de informações sensoreadas consiste no primeiro processo do mecanismo de simulação mental. Ele surge a partir do correlacionamento dos sensores de um sistema e o pacote *Raw*. Neste caso, os sensores recebem as informações dos objetos do ambiente e as repassam para as classes do pacote *Raw* para que possam criar a representação bruta do objeto em questão. Trata-se do tipo de representação mais elementar previsto no mecanismo. O processo recebe como entrada informações sensoreadas e retorna um objeto bruto, correspondendo àquele que foi percebido no mundo real e que é representado na mente do sistema. Um exemplo de representação bruta de um cubo vermelho que foi sensoreado seria: “cor: [255 (R), 0 (G), 0 (B)]; tamanho: [5 (X), 5 (Y), 5 (Z)]; forma: [6 (vértices)]”.

5.1.2.2 Sobre o Descobrimento de Categorias Ontológicas

A representação bruta dos objetos que foi gerada pelo processo de modelagem de informações sensoreadas servirá ao mecanismo de simulação mental de diferentes maneiras. Uma delas é aquela apresentada pelo processo de descobrir categorias ontológicas. Esse processo consiste na segmentação de um espaço conceitual e a definição de regiões semânticas que poderão depois categorizar a representação bruta de um objeto. Como vimos, na seção 2.5.1 (figura 3), essas regiões podem ser representadas por meio de protótipos (ou por objetos prototípicos, no caso de categorias de objetos). Apesar de protótipos serem representados por pontos, é importante lembrarmos que esses pontos na verdade representam toda uma região ao seu redor, de acordo com um diagrama de Voronoi. Por exemplo, uma representação bruta de um objeto que possui como atributos cor, tamanho e forma, pode dar origem às seguintes categorias:

- cor: [255 (R), 0 (G), 0 (B)] - protótipo no domínio da cor, representando uma região correspondente à categoria de propriedade “vermelho”;
- tamanho: [5 (X), 5 (Y), 5 (Z)] - protótipo no domínio do tamanho, representando uma região correspondente à categoria de propriedade “pequeno”;

- forma: [altura e largura] - protótipo no domínio da forma, representando uma região correspondente à categoria de propriedade “retângulo”;

A partir de um conjunto de protótipos representando diferentes categorias de propriedades, pode-se então compor objetos prototípicos, que poderiam ser usados em seguida para identificar categorias de objetos. O processo de descobrimento de categorias, portanto, consiste na reificação de diferentes tipos de categorias ontológicas que podem ser empregadas na tarefa de categorizar. Ele ocorre a partir do correlacionamento das classes do pacote *Raw*, *Conceptual* e *Ontology*. A Figura 9 apresenta um diagrama de atividades descrevendo esse processo.

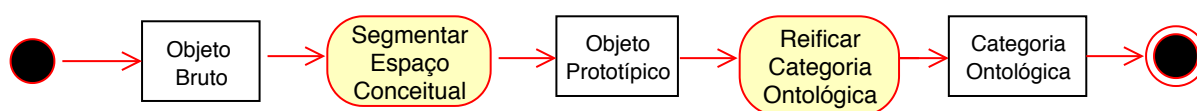


Figura 9 – Detalhamento do processo de descobrir categorias ontológicas.

Descobrir categorias ontológicas depende do emprego de técnicas de aprendizado de máquina, que ao comparar as representações brutas de um objeto a protótipos que representam espaços conceituais, desenvolve um processo de segmentação desse espaço, descobrindo novas regiões semânticas ou expandindo outras, dessa forma identificando novos domínios ou alterando as fronteiras de domínios já identificados. Um exemplo do funcionamento deste algoritmo pode ser visto na seção B.2 do Anexo B. O resultado da segmentação é um conjunto de objetos prototípicos que são usados para reificar categorias ontológicas e que, mais tarde, podem ser empregadas para categorizar outros objetos brutos. Sendo assim, esse processo recebe como entrada um objeto bruto e tem como saída uma categoria ontológica.

5.1.2.3 Sobre a Percepção de Eventos

Uma vez que se tenham categorias ontológicas identificadas para um processo de categorização, pode-se iniciar o processo de perceber eventos. Este processo ocorre utilizando-se diferentes elementos definidos nos pacotes *Raw*, *Ontology* e *Perception*. Ele consiste em categorizar um objeto que foi sensorado e identificar eventos que possam ter ocorrido com ele. A Figura 10 apresenta um diagrama de atividades descrevendo esse processo.

O processo de percepção de eventos inicia-se a partir da categorização de um objeto bruto que foi representado na mente do sistema. Neste ponto, o objeto bruto é categorizado, detectando-se suas propriedades por meio de sua comparação com os protótipos que representam as categorias. Por exemplo, a partir de um objeto bruto que possui três lados, detecta-se que ele pertence à categoria de objeto “triângulo”. Tanto o objeto

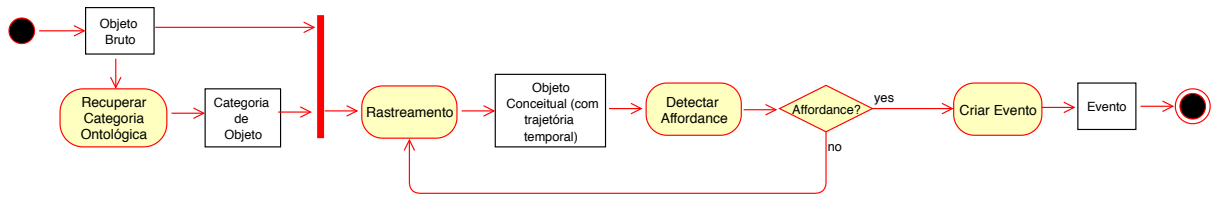


Figura 10 – Detalhamento do processo de perceber eventos.

bruto quanto sua categoria detectada (que podem variar ao longo do tempo) são mantidos na memória de curto prazo do sistema através de um algoritmo de **rastreamento**, que busca detectar mudanças que possam ter ocorrido com o transcorrimento do tempo. O objetivo dessa ação é montar a trajetória temporal do objeto em questão. O resultado desse subprocesso é a criação de um objeto conceitual, que permite diferenciá-lo dos demais objetos, inclusive daqueles que possuem a mesma categoria. A trajetória temporal do objeto conceitual é analisada a todo instante a fim de verificar se sobre ele foi aplicado uma *affordance*. Caso positivo, inicia-se a criação de um evento, que é o resultado final desse processo. Caso negativo, o rastreamento continua. A percepção de eventos consiste em um dos processos mais complexos do mecanismo de simulação. É a partir dele que o segundo nível de representação previsto no modelo ocorre (representação icônica) e, assim sendo, ele define a atuação do mecanismo de simulação mental no modo percepção. Por esse motivo, ele toma como entrada objetos brutos sensoreados e sua categoria de objeto e tem como saída um evento, que pode ser entendido como uma representação icônica de um fato que ocorreu no ambiente. Eventos normalmente são compostos por um **agente** (um objeto conceitual, o causador da *affordance*), um **paciente** (outro objeto conceitual, que teve sua propriedade alterada pela aplicação da *affordance*), uma ***affordance*** (a ação que promove a modificação de uma propriedade do paciente, neste caso, a ação de mover, que promove a modificação da propriedade posição), e seus eventuais **modificadores**.

5.1.2.4 Sobre a Nomeação de Eventos

Uma vez que eventos foram percebidos, eles podem ser empregados por diferentes processos cognitivos. Entretanto, como o objetivo desse trabalho é contribuir com o desenvolvimento da área de linguagem gramatical, as classes do pacote *Language* são empregadas para nomear as categorias ontológicas que compõem um evento. Como mostra a Figura 11, o processo de nomear eventos não é tão simples como parece.

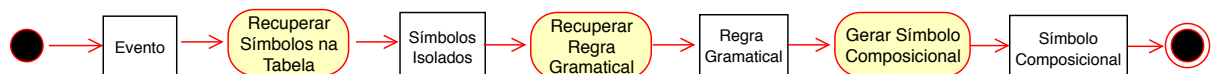


Figura 11 – Detalhamento do processo de nomear eventos.

O processo de nomeação inicia-se a partir de um evento que foi percebido pelo

mecanismo de simulação mental. Ele é repassado para o pacote de linguagem, onde as classes correspondentes vão iterar sobre cada categoria para recuperar um símbolo isolado para expressá-la. Caso exista mais um, um sorteio é feito para que apenas um símbolo possa ser escolhido. Como o intuito principal é formar um símbolo composicional, uma regra gramatical deve ser recuperada para combinar de maneira válida os símbolos recuperados. As regras previstas no modelo do mecanismo diz respeito ao posicionamento dos símbolos que representam o objeto conceitual, propriedade e modificador e ela é acordada entre os falantes de um idioma. Ao final, tem-se como resultado do processo um símbolo composicional que descreve linguisticamente o fato representado pelo evento. Dessa maneira, ele recebe como entrada um evento e tem como saída um símbolo composicional.

5.1.2.5 Sobre a Interpretação de Símbolos Composicionais

No sentido inverso ao processo de nomeação tem-se o processo de interpretação de símbolos composicionais. Ele consiste em traduzir um símbolo composicional no conjunto de percepções e sensações que o originou. Sendo assim, esse processo ocorre a partir do correlacionamento das classes que compõem os pacotes *Language*, *Ontology* e *Raw*. A Figura 12 apresenta o detalhamento deste processo.

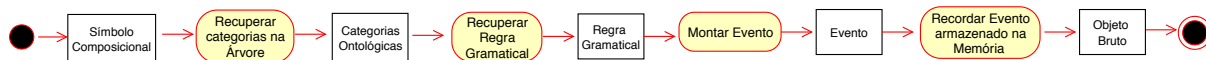


Figura 12 – Detalhamento do processo de interpretar símbolos composicionais.

O processo de interpretação inicia-se a partir de um símbolo composicional recebido. O símbolo composicional é segmentado nos símbolos isolados que o compõem e a árvore de símbolos é consultada para recuperar, para cada símbolo identificado, a categoria ontológica que faz sua fundamentação. No caso de haver mais de uma categoria, um sorteio é realizado. De posse das categorias, é necessário recuperar a regra gramatical que combine as categorias de maneira válida, a fim de entender a forma como o evento ocorreu. Ao contrário do processo de nomeação, a regra gramatical aqui faz sentido para que se possa montar corretamente o evento da interpretação. No caso de existir mais de uma regra para combinar as categorias, aquela que tiver maior grau de confiança é eleita. A partir do evento montado, a memória de longo prazo é acessada em busca de eventos do passado que envolvam o objeto bruto em questão. Acessar a memória em busca desse objeto consiste no reviver mental do trecho da experiência que deu origem ao evento, sendo possível sentir o conjunto de sensações e percepções que deram origem a ele. O reviver mental consiste na interpretação do evento. Esse processo, portanto, recebe como entrada um símbolo composicional e tem como saída um objeto bruto recordado que permite correlacionar o símbolo composicional com o evento que ele descreve. O processo de interpretação, juntamente com o processo de nomeação, consistem no mecanismo de simulação mental modo linguagem.

Vale ressaltar que, além dos processos de nomeação e interpretação, o mecanismo de simulação mental no modo linguagem também envolve um conjunto de estratégias que permitem que uma comunidade de falantes possa emergir e evoluir uma língua comum entre si. Em virtude do número elevado de recursos que foram desenvolvidos para cumprir todas as funcionalidades desta parte do mecanismo, optamos por apresentá-lo em um capítulo separado. Assim sendo, no Capítulo 6 apresentaremos mais detalhes a respeito do mecanismo de simulação mental no modo linguagem e na próxima seção apresentaremos os detalhes do mecanismo de simulação mental no modo percepção.

5.2 Simulação Mental: Modo Percepção

A simulação mental modo percepção tem como objetivo rastrear alterações espaço-temporais dos objetos, classificá-las utilizando as categorias previstas no pacote *Ontology* e detectar a ocorrência de eventos. Ela consiste no processo de perceber eventos e é a partir dela que a representação icônica da realidade é construída. Para entender seu funcionamento, vejamos um exemplo a seguir.

A Figura 13 apresenta uma cena na qual existe um conjunto de objetos, que pode ter uma ou mais de suas propriedades modificadas ao longo do tempo.

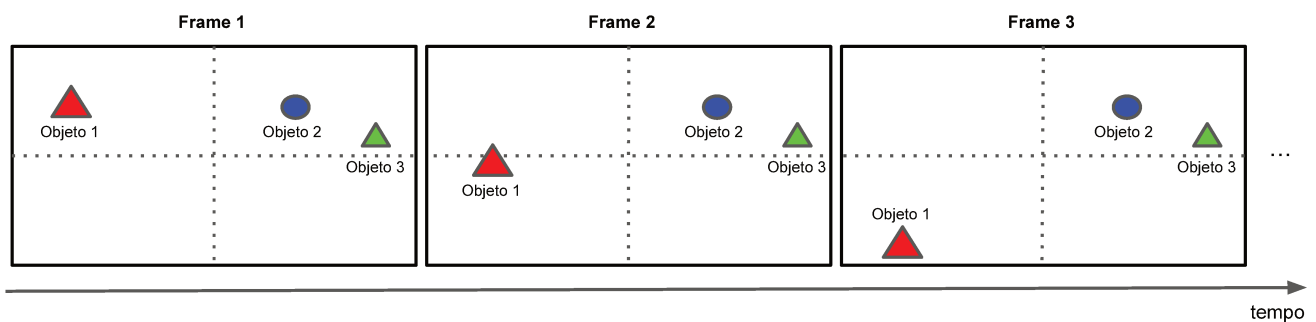


Figura 13 – O sensoramento de vários objetos ao longo do tempo e suas transformações espaço-temporal.

O tempo no *framework* é medido em quadros ou *frames*. Analisando esses objetos *frame a frame*, pode-se concluir que o objeto 1 sofreu uma alteração em sua propriedade posição e com isso um evento pode ser detectado. Eventos são compostos por um **agente** (um objeto conceitual, o causador da *affordance*), um **paciente** (outro objeto conceitual, que teve sua propriedade alterada pela aplicação da *affordance*, - neste caso, o próprio Objeto 1), uma ***affordance*** (a ação que promove a modificação de uma propriedade do paciente, neste caso, a ação de mover, que promove a modificação da propriedade posição), e seus eventuais **modificadores** (que poderiam indicar alguma informação adicional ou restrição sobre a modificação da propriedade atingida, como por exemplo, “rapidamente”, ou “para baixo”).

Para ser capaz de perceber eventos dessa natureza o mecanismo de simulação mental conta com um algoritmo de rastreamento de objetos, que será apresentado em detalhes a seguir.

5.2.1 Algoritmo de Rastreamento

Rastrear objetos estáticos é uma tarefa relativamente simples. A complexidade é maior quando esses objetos sofrem alterações em suas propriedades e partes constituintes, ao longo do tempo, onde as seguintes situações podem ocorrer: (i) entre um *frame* e outro, objetos podem ter desaparecido e reaparecido; (ii) os objetos podem estar parcialmente oclusos, ou ainda, podem ter sofrido uma transformação que alterou totalmente a configuração de suas propriedades e partes. Sendo assim, é preciso rastrear esses objetos ao longo de diversos *frames* para que seja possível detectar essas mudanças e, consequentemente, os eventos ocorridos.

Antes de apresentar a implementação do algoritmo, se faz necessário explicar as estruturas que oferecem suporte para o seu funcionamento. Elas são: a **memória imediata (também referenciada como *buffer* imediato)** e a **memória de curto prazo**.

A memória imediata (ou *buffer* imediato), implementada na classe **ImmediatePath**, é uma memória de curtíssimo prazo e de tamanho limitado, cuja função consiste em auxiliar a percepção durante a tarefa de rastreamento dos objetos, a partir de suas representações brutas. Para isso, a cada novo *frame*, enumeramos todas as possibilidades de interpretação dos objetos sensoreados, os adicionando como passos (**steps**) de continuação de caminhos (**paths**). Cada caminho indica uma possível trajetória de um objeto conceitual nos últimos *frames*. Ao armazenar caminhos alternativos para um mesmo objeto, é possível considerar, além da similaridade da representação do objeto em um *frame* com suas representações nos *frames* anteriores, também a possibilidade de sua correta continuação nos *frames* seguintes.

Já a memória de curto prazo armazena cada objeto conceitual identificado e suas representações brutas, nos diferentes instantes de tempo em que foram percebidos. Ela retém as informações por um período pequeno de tempo, o suficiente para verificar se alguma *affordance* foi aplicada e, caso positivo, detectar um evento, que é registrado permanentemente na memória de longo prazo. O exemplo apresentado na Figura 14 ajuda a entender como ambas as estruturas são usadas no algoritmo implementado.

Na linha do tempo, representamos uma cena em dois *frames* distintos, sendo o *frame* 1 o mais antigo (denominado T1) e o *frame* 2 o mais recente (denominado T2). O primeiro *frame* a ser processado é T1, onde três representações brutas de objetos (a1, a2 e a3, instâncias da classe **RawObject**) são sensoreadas. Seguindo a sequência do processo

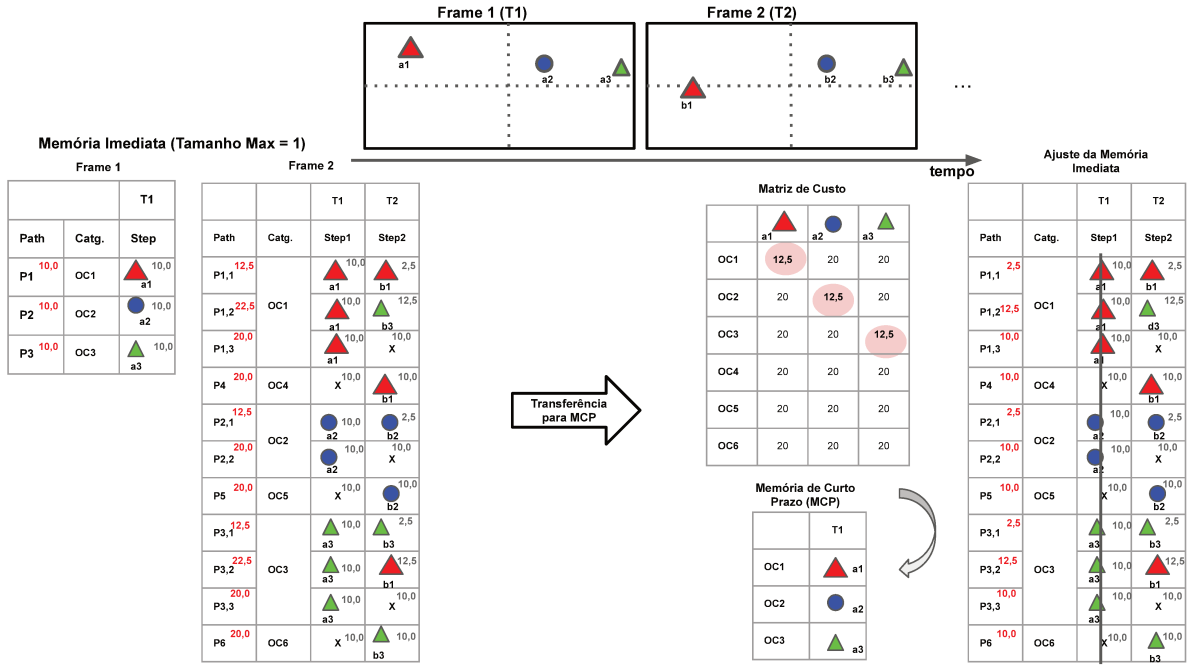


Figura 14 – Esquema do Algoritmo de Rastreamento.

de perceber eventos, suas propriedades são calculadas, para identificar suas categorias de objeto. O próximo passo é o rastreamento de cada objeto.

T1 é o primeiro *frame* e ainda não existem *paths* na memória, portanto nenhuma continuação é necessária. Para cada representação bruta sensoreada, um novo *path* (P1, P2 e P3) é criado, com um *step* contendo a representação, e um objeto conceitual construído a partir da mesma (OC1, OC2 e OC3) é associado a ele.

O filtro de *Kalman* (**UnscentedKalmanFilter**) é utilizado para remover ruídos dos valores brutos sensoreados, e em *frames* posteriores, também é utilizado para construir representações brutas estimadas para a continuação dos *paths*. Estas previsões são utilizadas no cálculo do custo do *step* que continuará os *paths* existentes, com novas representações brutas sensoreadas.

Como a memória imediata estava vazia e ainda não havia atingido sua capacidade máxima (1 *frame*), todos os três *paths* criados são armazenados na mesma, como mostra a tabela *Frame 1* da Figura 14. Repare na tabela, que um *path* é uma associação entre um objeto conceitual (cuja sigla se inicia com **OC**) e suas possíveis representações brutas (**steps**), ao longo do tempo. Como o filtro de *Kalman* não fez nenhuma previsão, o custo do *step* é um valor fixo de inicialização, neste caso, 10. Neste ponto, o fluxo da simulação mental do *frame 1* é finalizado, pois a memória imediata ainda não está acima de sua capacidade.

Iniciando um novo fluxo, o *frame 2* é processado. Três novas representações brutas (b1, b2 e b3) são sensoreadas, e o algoritmo de rastreamento deve identificar se são

continuações de *paths* anteriores ou novos objetos sendo sensoreados pela primeira vez. Então, suas propriedades são calculadas e suas categorias de objeto identificadas. Neste ponto, três situações são consideradas:

1. As representações brutas sensoreadas em T2 podem ser continuação dos *paths* iniciados anteriormente. Uma representação bruta só é considerada como uma possível continuação de um *path* se representar um objeto conceitual de mesma categoria do objeto conceitual ao qual o *path* está associado. A cada possível continuação, um “custo de continuação” é associado, calculado a partir da diferença entre a representação bruta e a estimativa produzida pelo filtro de *Kalman*. A seguir, as possíveis continuações são submetidas aos seguintes filtros:
 - a) o custo do *step* está entre os menores para continuação do *path* (utiliza-se um limiar de corte calculado pela classe **CostThreshold** sobre os custos das continuações do *path*);
 - b) o *path* atual está entre os mais compatíveis com a representação bruta sensoreada (utiliza-se um limiar de corte calculado pela classe **CostThreshold** sobre os custos de continuações utilizando a representação bruta).
2. Os objetos brutos sensoreados em T1 podem estar totalmente oclusos em T2. Se assim for, a melhor opção é continuar todos os *paths* iniciados em T1 com a oclusão (na figura demarcada com o sinal de “X”). Vale ressaltar que o custo de um *step* com oclusão é sempre um valor alto e fixo. No exemplo apresentado este valor é 10.
3. Os objetos brutos sensoreados em T2 podem ter estado oclusos em T1. Sendo assim, outra atitude a se tomar é criar *paths* para novos objetos conceituais (OC4, OC5 e OC6). É incluído um *step* de oclusão nesses *paths* em T1, e em T2 é incluído um *step* com a representação bruta sensoreada.

Após a atualização dos *paths* existentes e a inclusão dos novos *paths*, a memória imediata passa a conter dois *frames*, excedendo, assim, o limite de um *frame*. Portanto, deve-se iniciar um novo processo que consiste na transferência das informações da memória imediata para a memória de curto prazo. Esse processo envolve a escolha da melhor representação bruta para cada objeto conceitual. À direita da Figura 14 há um esboço de como esse processo ocorre.

A partir do tempo mais antigo da memória imediata (T1), constrói-se uma matriz de custo, que irá correlacionar cada objeto conceitual identificado (OC1, OC2, OC3, OC4, OC5 e OC6) com todas as possíveis representações brutas que podem representá-lo naquele instante de tempo. O menor custo entre os *paths* de cada associação será usado como um “custo da associação”.

Na Tabela *Frame 2* da Figura 14, existem três *paths* (P1,1; P1,2 e P1,3) que utilizam *a1* para representar, em T1, o objeto conceitual OC1. O custo do *path* P1,1, que é o menor entre os três, é utilizado na célula correspondente da matriz de custo. Caso uma associação não possua um *path*, adota-se na matriz um valor alto (no caso, 20), para que aquela associação não seja escolhida. Foi o que aconteceu na associação entre o objeto conceitual OC1 e as representações *a2* e *a3*.

Com a matriz de custo toda preenchida, um algoritmo de atribuição¹ é utilizado para decidir quais associações minimizam o custo total. Duas representações brutas diferentes nunca são escolhidas para um mesmo objeto conceitual e dois objetos conceituais diferentes nunca são representados por uma mesma representação bruta. As associações escolhidas foram demarcadas na matriz com o sombreado vermelho. Ao final do algoritmo, as representações brutas escolhidas são transferidas para a memória de curto prazo, associadas aos seus respectivos objetos conceituais.

Após a escolha das representações brutas, removemos da memória imediata os *paths* que possuem representações brutas diferentes da escolhida, para seu objeto conceitual. Além disso, também removemos dos *paths* da memória imediata os *steps* que contêm representações brutas no tempo antigo, a ser removido. No exemplo da Figura 14 não existiam *paths* que começavam com representações brutas diferentes das escolhidas, portanto só removemos os *steps* em T1 e mantivemos todos os *paths*.

A continuação do rastreamento iniciado no exemplo da Figura 14 pode ser vista na Figura 15.

Desta vez, novas representações brutas são sensoreadas em T3 e o processo descrito anteriormente é repetido para escolher representações brutas para os objetos conceituais em T2. Vale ressaltar que parte da figura foi oculta, por razões de formatação e espaço.

Repare que com a inserção das informações sensoreadas no *frame 3*, a memória imediata atinge seu tamanho máximo e novamente deve ocorrer a transferência de informações para a memória de curto prazo. Entre todos os *paths* criados para representar um objeto conceitual, o que possui menor custo é eleito para montar a matriz de custo. Novamente, caso uma associação não possua um *path*, adota-se na matriz um valor alto (no caso, 20), para que aquela associação não seja escolhida. O algoritmo de atribuição é utilizado para decidir quais associações minimizam o custo total. As representações brutas eleitas (demarcadas com sombreado vermelho) são transferidas para memória de curto prazo.

No exemplo da Figura 15 o ajuste da memória imediata começa pela identificação e remoção dos *paths* que possuem representações brutas diferentes da escolhida para

¹ Foi utilizado o algoritmo de emparelhamento perfeito com custo mínimo de *Kuhn–Munkres* introduzido em *Munkres (1957)*, cuja implementação está na classe **CostMatrix**.

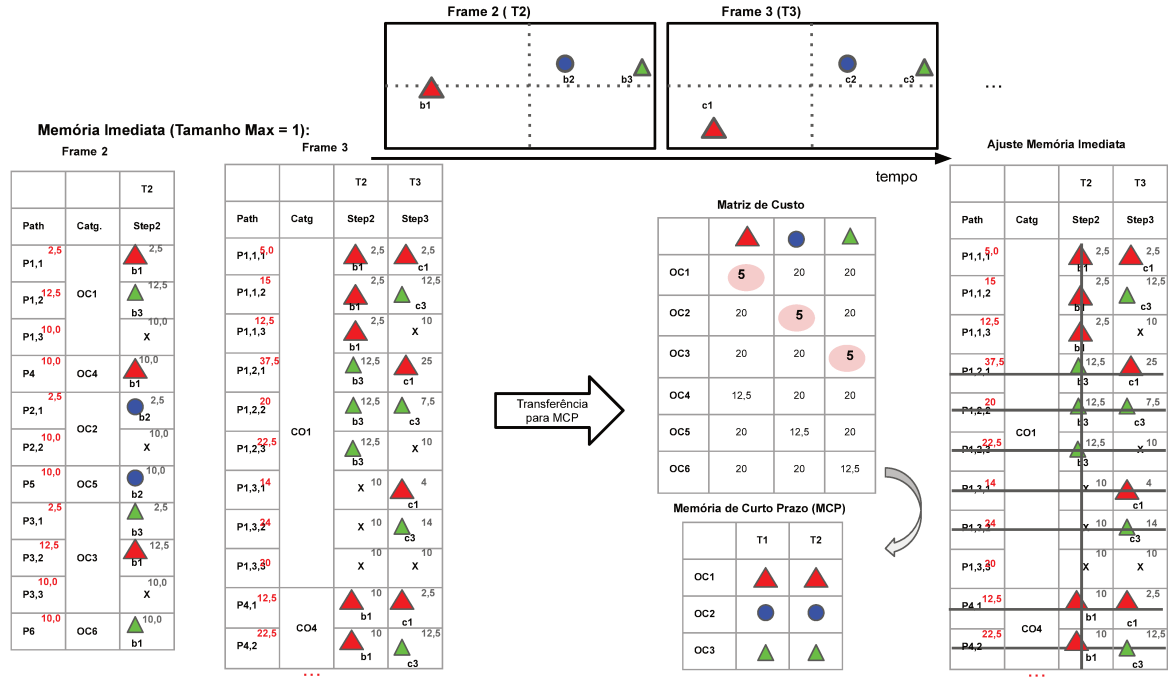


Figura 15 – Esquema do Algoritmo de Rastreamento - Continuação do esquema apresentado na Figura 14.

seu objeto conceitual. Repare que isso aconteceu com os *paths* (P1,2,1; P1,2,2; P1,2,3; P1,3,1; P1,3,2; P4,1; P4,2). Logo em seguida, os *paths* são atualizados com a remoção dos *steps* que contém representações brutas do tempo anterior.

À medida que novas representações brutas são transferidas para a memória de curto prazo, as demais etapas do fluxo da simulação mental (modo percepção) podem ocorrer. Continuando a partir da etapa 4, estas representações são analisadas pelo método **detect()** para verificar se alguma *affordance* foi aplicada e detectar seus eventuais modificadores. Neste caso, é construído um evento (composto pelos objetos conceituais analisados, a *affordance* detectada e seus modificadores). O evento e as informações brutas que o originaram são armazenados na memória de longo prazo. Caso existam episódios ativos, com foco nos objetos conceituais em questão, o evento retornado é também adicionado aos mesmos.

Vale ressaltar que todas as categorias ontológicas identificadas (propriedades, categorias de objetos, objetos conceituais, *affordances*, modificadores, eventos e episódios) durante o fluxo são representações icônicas da realidade e podem ser usadas em outros processos cognitivos.

5.3 Considerações Finais

Para a criação de símbolos gramaticais fundamentados, partimos da hipótese de que é necessária a criação de um tipo de representação intermediária que, durante o processo de categorização, segmente a realidade em termos de categorias que possam ser empregadas tanto para a criação de estruturas simbólicas gramaticais, quanto para sua interpretação semântica. O propósito do mecanismo de simulação mental apresentado neste capítulo almejou a validação dessa hipótese de pesquisa.

Um de seus modos de atuação é como processo perceptual, cujo objetivo é criar a infraestrutura necessária para que conceitos possam ser aprendidos e, no contexto deste trabalho, serem empregados para fundamentar o significado de símbolos de alta-ordem. Assim, ele executa dois processos distintos, mas que são correlacionadas. Um, responsável por sensoriar objetos, utilizar suas informações brutas para segmentar o espaço conceitual e mapear suas regiões semânticas em categorias ontológicas. Já a outra, responsável por rastrear objetos e suas transformações, categorizando-os por meio das categorias ontológicas aprendidas.

Para garantir a categorização dos eventos (a estrutura mais complexa do pacote *Ontology*), o mecanismo conta com um *buffer* imediato e duas memórias, de curto e longo prazo. Essas estruturas de dados são de vital importância para seu funcionamento, pois além de permitir o rastreamento correto do objeto ao longo dos diversos *frames* que descrevem as cenas do mundo, as informações que elas mantêm ajudarão na interpretação simbólica, como veremos no capítulo 6. Dessa maneira, também conferimos ao mecanismo a capacidade de atuar como um sistema de símbolos perceptuais, como aquele proposto por Barsalou (1999), ao permitir que o conteúdo das memórias possa ser restaurado e vivenciado durante a interpretação simbólica. Sendo assim, a aplicação do mecanismo de simulação mental, bem como de todas as funcionalidades mencionadas ao longo do texto, poderão ser experimentadas no capítulo 7. Espera-se que ao estabelecer atos comunicativos usando esse *framework* como infraestrutura, duas características sejam alcançadas:

1. Símbolos de alta-ordem possam emergir, sem que seu significado dependa totalmente do projetista do sistema.
2. Garantir a interoperabilidade semântica entre os interlocutores diversos dos atos comunicativos estabelecidos.

6. MECANISMO DE SIMULAÇÃO MENTAL: MODO LINGUAGEM

Neste capítulo apresentamos o mecanismo de simulação mental, atuando no modo de linguagem. Além de permitir a geração e interpretação de símbolos gramaticais, ele também disponibiliza regras de produção para definição de uma gramática, ainda que de escopo simplificado. As regras atuam como mecanismo de evolução, de modo que, símbolos de alta-ordem, a função de cada símbolo isoladamente (papel gramatical) e a função dos símbolos na sentença (papel sintático) possam ser aprendidos e, assim, definir uma proto-linguagem¹.

Entende-se como um símbolo de alta-ordem tanto um símbolo isolado como uma sentença. Uma sentença é constituída por uma sequência de um ou mais símbolos isolados², que em conjunto exprimem um significado envolvendo ideias, emoções, ordens, apelos, eventos e outros. Em uma sentença, dizemos que cada símbolo possui três funções distintas: gramatical, sintática e semântica. Um exemplo de sentença que pode ser gerada pelo pacote de linguagem proposto pode ser observado na Figura 16.

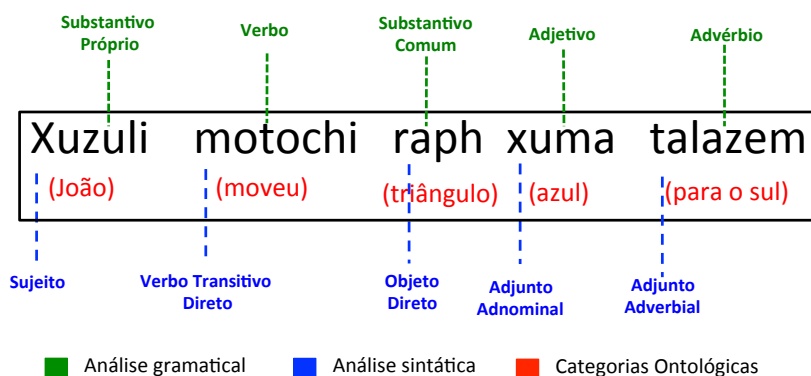


Figura 16 – Análise gramatical e sintática de uma sentença que pode ser gerada pelo pacote de linguagem proposto.

A sentença apresentada expressa um evento onde um indivíduo movimenta um objeto para uma direção específica (significado: “João moveu triângulo azul para o sul”). Analisando a função gramatical de cada símbolo (destacada na cor verde), percebemos que ela é formada por: dois substantivos (um próprio e outro comum), um verbo, um adjetivo e um advérbio. Já na estrutura da sentença, cada símbolo possui uma função sintática diferente (destacada na cor azul) que depende do relacionamento com os outros

¹ Segundo Bickerton (2009), pode-se entender uma proto-linguagem como uma linguagem simplificada que é empregada antes do surgimento de uma linguagem completa.

² Existem diversas sentenças com um único símbolo isolado, tais como "Pare." ou "Cuidado!", embora a grande maioria das sentenças tenha dois símbolos ou mais.

símbolos. A função semântica (associada a uma categoria ontológica) de cada símbolo é destacada na cor vermelha.

O mecanismo de evolução proposto não descreve o significado dos símbolos manipulados ou muito menos o que pode ser feito com eles em um determinado contexto. Ele contém os métodos que irão determinar seu formato, especialmente das sentenças. Neste caso, a fundamentação simbólica ocorre por meio do reviver das informações brutas que deram origem às categorias ontológicas detectadas durante o processo perceptual da simulação mental. Daí a justificativa de acoplamento entre ambos os pacotes de software. Enquanto o capítulo anterior apresentou a simulação mental no modo percepção, este apresentará a simulação mental no modo linguagem.

A seguir apresentamos mais detalhes do mecanismo de linguagem, destacando suas principais funcionalidades.

6.1 Como Símbolos Isolados são Aprendidos e Associados às suas Funções Gramaticais?

A criação de símbolos isolados (com fundamentação semântica e funções gramaticais), tem relação direta com o processo perceptual, mais especificamente com a identificação de categorias ontológicas durante esse processo. Cada categoria tem uma função específica no pacote de linguagem. Assim, podemos afirmar que:

- Uma **categoria de objeto** representa conceitualmente a funcionalidade de **substantivos comuns**.
- Um **objeto conceitual** representa conceitualmente a funcionalidade de **substantivos próprios**.
- Uma **propriedade** representa conceitualmente a funcionalidade de **adjetivos**.
- Uma **affordance** representa conceitualmente a funcionalidade de **verbos**.
- Um **modificador** representa conceitualmente a funcionalidade de **advérbios**.
- Um **evento** representa uma **sentença verbal** e surgirá sempre que uma *affordance* for aplicada.
- Um **episódio** representa um pequeno trecho de um **texto** que descreve a história de um objeto específico.

Assim, toda vez que uma nova categoria é instanciada durante o processo perceptual e houver necessidade, um símbolo pode ser criado e sua função gramatical

estará diretamente relacionada à categoria ontológica que ele pertence. Para isso, a classe **Language** disponibiliza os métodos **createSymbol()** e **registerSymbol()**.

O método **createSymbol()** simplesmente cria um símbolo para a categoria que é informada como parâmetro. Já o símbolo criado, juntamente com sua categoria, são repassados para o método **registerSymbol()**, que irá registrar essa associação nas estruturas de dados mantidas pelo pacote de linguagem. Falaremos sobre o mecanismo de aprendizado gramatical subjacente a esse método mais adiante.

6.1.1 Sobre as Estruturas de Dados Mantidas pelo Mecanismo de Linguagem

Como os símbolos oriundos do mecanismo de linguagem são trocados através de atos comunicativos (aqui chamados de jogos de linguagem), se faz necessário mantê-los em estruturas de dados correlacionadas às categorias usadas para modelar seu significado. Durante a evolução de um jogo de linguagem, os agentes devem ser capazes de representar casos de ambiguidade (símbolos que têm mais de um significado) e sinonímia (categorias que possuem mais de um símbolo para referenciá-las), pelo menos provisoriamente, até que se dê a convergência semântica entre um grupo de interlocutores, que permita a interoperabilidade entre eles. Sendo assim, por questão de implementação, o mecanismo de linguagem mantém duas estruturas de dados principais: uma **árvore de símbolos** e uma **tabela de categorias**. Embora as duas tenham a mesma finalidade, que é manter pares (símbolo, categoria), elas são usadas em momentos distintos. Enquanto a árvore facilita a descoberta de casos de ambiguidade, a tabela facilita a descoberta de casos de sinonímia.

A árvore armazena a associação de um símbolo com suas categorias, registrando ainda a probabilidade de uma categoria fazer a fundamentação simbólica de um símbolo. Por esse motivo, ela é usada pelo método **symbolToElement()** da classe **Language** durante a interpretação simbólica. Este método converte qualquer símbolo, seja ele um símbolo isolado ou uma sentença (uma combinação destes), em sua(s) respectiva(s) categoria(s) ontológica(s). Observe que, em caso de ambiguidade, um mesmo símbolo poderá estar associado a mais de uma categoria. O método retornará portanto todas as possíveis categorias relacionadas ao símbolo. Já a tabela armazena a associação de uma categoria ontológica com seus possíveis símbolos. Além da probabilidade de um símbolo expressar a categoria em questão, há também a probabilidade do agente exercer sua inventividade³, e dessa forma resolver propor um novo símbolo para a categoria. Por esse motivo, ela será usada pelo método **elementToSymbol()** da classe **Language** durante a geração simbólica. Este método traduz um ou vários elementos da categoria ontológica

³ A inventividade corresponde à criação de um novo símbolo para nomear uma categoria quando por algum motivo criou-se uma associação equivocada anteriormente.

nos seus símbolos correspondentes. A Figura 17 ilustra ambas as estruturas de dados citadas. A probabilidade da inventividade na figura está indicada por “#i”.

Supondo que ambas as estruturas de dados estivessem vazias e fossem armazenar os símbolos da sentença: “**xuzuli motochi raph xuma talazem**” (cujo significado é: “João moveu triângulo azul para o sul”), elas ficariam como mostra a Figura 17.

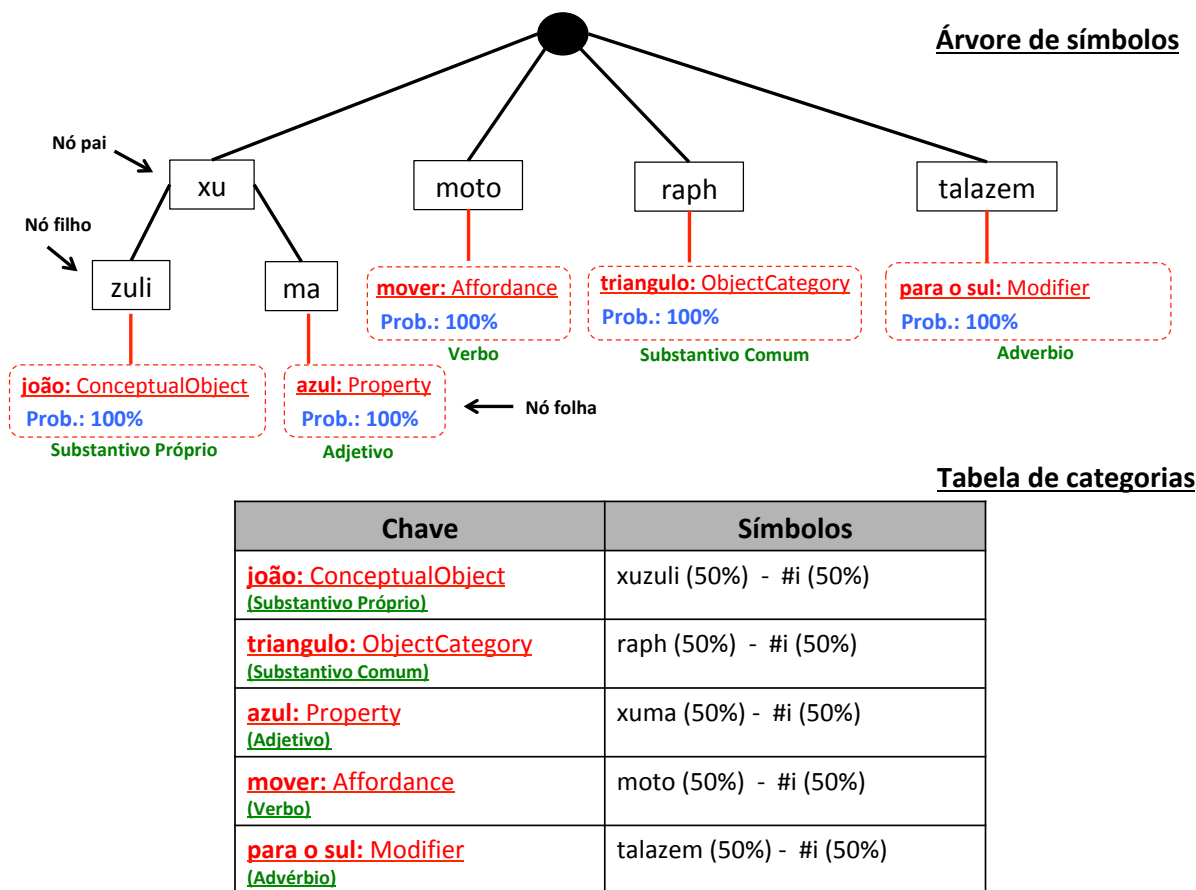


Figura 17 – As estruturas de dados mantidas pelo mecanismo de linguagem: a Árvore de Símbolos e a Tabela de Categorias. Ambas preenchidas com os símbolos da sentença “xuzuli motochi raph xuma talazem”.

Algumas observações devem ser feitas aqui:

1. Os nós intermediários da árvore de símbolos (nós pais e filhos) armazenam os símbolos, decompostos em termos das partes que se repetem em diferentes símbolos já conhecidos. Os nós folhas guardam as categorias que o símbolo representa, junto com sua probabilidade de representação. Já a tabela associa uma categoria com os possíveis símbolos que podem designá-la e suas probabilidades.
2. No caso da árvore, os símbolos “xuzuli” e “xuma” iniciam com a mesma sílaba. Por esse motivo eles foram divididos, compartilhando o nó que contém a sílaba “xu”. Suas demais sílabas (“zuli” e “ma”) ficam em nós distintos.

3. O símbolo “motochi” designa um verbo que está flexionado no passado (no exemplo isso é indicado pela terminação “chi”). Como as estruturas de dados guardam símbolos sem nenhum tipo de flexão, a terminação “chi” foi retirada dos mesmos. Vale ressaltar que essas flexões são adicionadas no símbolo no momento em que ele será expresso.
4. Na tabela de categorias aparece o ícone “#i” que designa a capacidade de inventar um novo símbolo para nomear uma categoria. Assim, ao tentar nomear uma categoria haverá a possibilidade de inventar um novo nome para ela ou recuperar aqueles já registrados (caso existam). Essa habilidade foi implementada para conferir ao sistema a capacidade de criar outros nomes quando uma associação equivocada ocorrer. Porém, quanto mais nomes forem registrados para uma categoria, menor será a capacidade de inventar. Ao longo do tempo, essa habilidade quase nunca será utilizada. Esse ajuste dependerá dos acertos linguísticos que um símbolo levará em um diálogo e será realizado de acordo com uma estratégia de aprendizado gramatical implementado.

6.1.2 Sobre o Aprendizado Gramatical

Como pode ser visto na Figura 17, tanto categorias (folhas da tabela de símbolos) como símbolos (coluna da direita da tabela de categorias) possuem um valor numérico atrelado a sua estrutura, que diz respeito à probabilidade de uma categoria fundamentar um símbolo e de um símbolo expressar uma categoria. Esses valores são atualizados a partir de estratégias de aprendizado empregadas enquanto ocorre a competição entre os elementos.

Como apresentado no capítulo 3, Wellens (2012) empregou três estratégias diferentes nos jogos de linguagem que ele modelou. Elas serviram de inspiração para a proposição de outras estratégias que são utilizadas nesse trabalho. As estratégias foram implementadas no método `registerSymbol()` da classe `Language` e modelam um mecanismo associativo entre símbolos e categorias, garantindo, assim, o aprendizado gramatical. Tal mecanismo de aprendizado conta o número de vezes em que os pares (símbolo, categoria) e (categoria, símbolo) são apresentados ao sistema e aumenta ou diminui a probabilidade de associação entre cada par, em suas respectivas estruturas de dados, de modo que a soma das probabilidades fique sempre em torno de 100%. Entretanto, como as estruturas mantêm as mesmas informações, porém de maneira diferente, um mecanismo de atualização eficiente se faz necessário para manter a coerência entre elas. Sendo assim, uma estratégia de reforço e penalização das associações foi proposta.

- Cada categoria possui um contador c_c que registra quantos símbolos já foram refor-

çados em associação à categoria. Inicialmente, c_c tem valor 1, indicando que, mesmo que não tenha nenhum símbolo ainda associado, existe pelo menos a capacidade de “inventar símbolos”.

- Cada símbolo possui um contador c_s que registra quantas categorias já foram reforçadas em associação ao símbolo. Inicialmente, c_s tem valor 0, representando que o símbolo atual ainda não foi associado a nenhum elemento.
- Chamamos de p_a a taxa de incremento atual, que pode ser configurada a critério do projetista. Para os exemplos apresentados a seguir, ela vale 1 (significando 100%).
- Chamamos de p_c e p_s as taxas de correção, que serão calculadas de acordo com os valores atuais das variáveis correlacionadas.
- Chamamos de p_{mr} o percentual mínimo de reforço, que indica o mínimo valor a ser adicionado a uma nova associação.
- Chamamos de p_{mp} o percentual mínimo de penalização, que indica o mínimo valor que uma associação existente sendo penalizada perderá.
- Chamamos de p_{rem} o percentual de remoção, que indica o percentual mínimo que uma associação deve ter para não ser removida das estruturas de dados.
- Chamamos de p_{ne} o percentual utilizado durante o reforço de uma associação, para reduzir a penalização aplicada às demais associações da categoria e do símbolo em questão.

As estratégias adotadas foram:

Estratégia para reforço de associações da árvore e tabela

1. Se a associação já existir, somamos $p_{mr} + (p_a - p_{mr})/c_c$ na tabela e $p_{mr} + (p_a - p_{mr})/c_s$ na árvore.
2. Se a associação não existir, inserimos com $p_{mr} + (p_a - p_{mr})/c_c$ na tabela e com $p_{mr} + (p_a - p_{mr})/c_s$ na árvore. Caso $c_s = 0$, inserimos com 100% na árvore.
3. Multiplicamos todas as associações por $p_c = 1/(1 + p_{mr} + (p_a - p_{mr})/c_c)$ na tabela e por $p_s = 1/(1 + p_{mr} + (p_a - p_{mr})/c_s)$ na árvore. Caso $c_s = 0$, usamos $p_s=100\%$.
4. Os contadores do elemento e do símbolo são incrementados com p_a ;
5. Penalizamos as demais categorias do símbolo na tabela de categorias, e penalizamos os demais símbolos da categoria na árvore de símbolos. A porcentagem dessas

penalizações é reduzida pelo parâmetro p_{ne} , sendo menos significativa que aquela aplicada em uma penalização direta.

Estratégia para penalização das associações da árvore e tabela

1. Calculamos as probabilidades $p_c = 1 + p_{mp} + (p_a - p_{mp})/c_c$ e $p_s = 1 + p_{mp} + (p_a - p_{mp})/c_s$.
2. Multiplicamos as probabilidades de todas as associações existentes por p_c na tabela de categorias e por p_s na árvore de símbolos.
3. Se a penalidade não for no “inventar símbolo”, subtraímos da probabilidade da associação sendo penalizada $p_c - 1$ na tabela de categorias e $p_s - 1$ na árvore de símbolos.
4. Se a probabilidade da associação sendo penalizada se tornar menor ou igual a 0, a associação é removida da tabela de elementos e/ou da árvore de símbolos, e as demais probabilidades são ajustadas para que a soma continue resultando em 100%.

Um exemplo de funcionamento de ambas as estratégias pode ser visto na Figura 18.

Observando a árvore de símbolos apresentada anteriormente, somente a propriedade “azul” fazia a representação simbólica da palavra “xuma”. Supondo que a categoria de objeto “quadrado” também passe a fundamentar seu significado, podemos observar que a categoria “azul” de 100% de probabilidade passou a ter 50% e a categoria “quadrado” 50% também. Neste caso, houve uma alteração na probabilidade, pois há duas categorias distintas disputando para fundamentar o significado do mesmo símbolo. Por outro lado, na tabela de categorias uma nova linha foi adicionada para manter a probabilidade do símbolo “xuma” descrever a categoria “quadrado”. Como se trata de uma nova categoria, o símbolo de inventar (#i) e o símbolo “xuma” disputam entre si para representá-la, tendo os dois 50% de chances de serem eleitos em uma próxima oportunidade. Como a categoria azul também era designada pelo símbolo “xuma”, ele será penalizado, passando a ter 25% de probabilidade de ser eleito e a capacidade de inventar, por consequência, recebeu um acréscimo de sua probabilidade, indo para 75%.

Um outro exemplo, um pouco mais sofisticado, é mostrado na Figura 19.

Neste caso, a palavra “ha” foi empregada para também designar a categoria “para o sul”. Sendo assim, ela entra na árvore de símbolos com 100% de probabilidade, pois se trata de uma nova associação. O próximo passo seria penalizar os símbolos das outras associações e que também são empregados para designar a mesma categoria. Como neste caso há apenas a ocorrência do símbolo “talazem”, sua probabilidade permanecerá

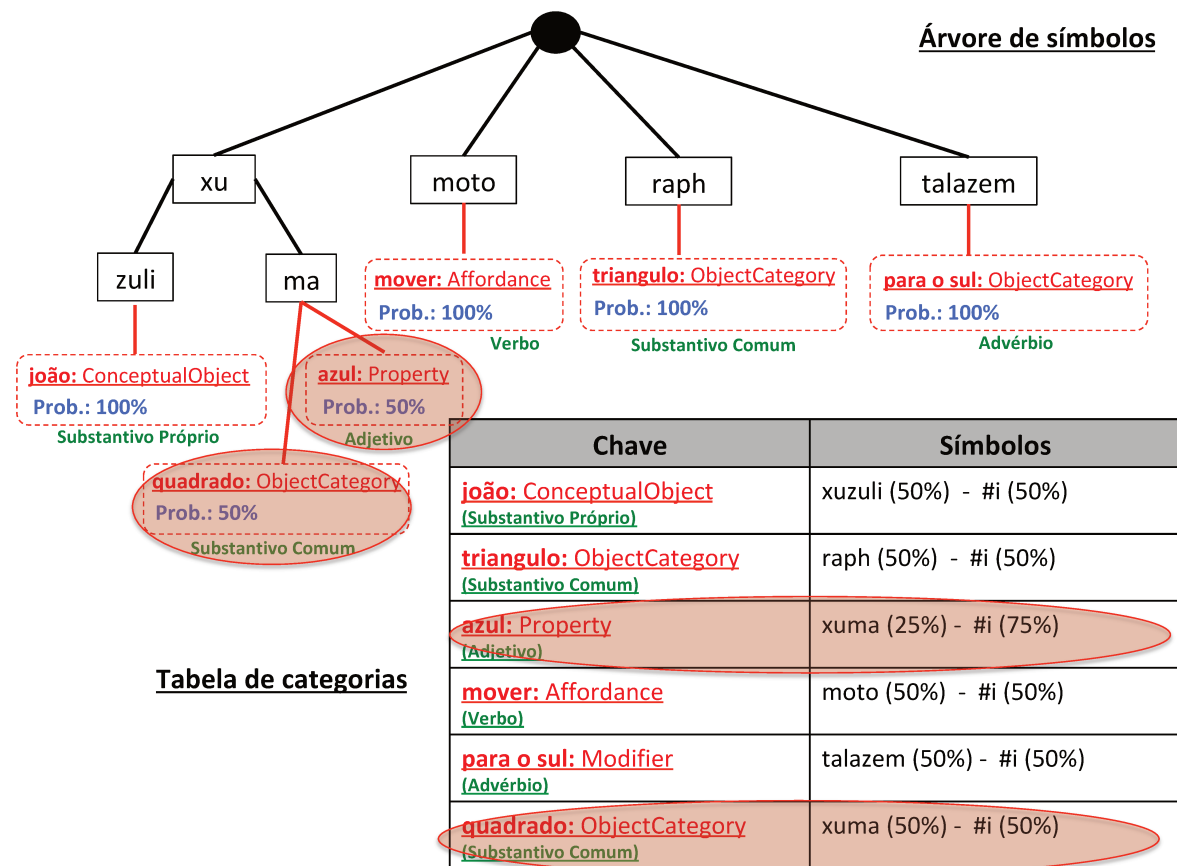


Figura 18 – Exemplo 1 - Ajuste do percentual de associação entre símbolo + categorias na árvore de símbolos e entre categoria + símbolos na tabela de categorias.

intacta em 100%. Já na tabela de categorias, na linha cuja chave é a categoria “para o sul”, há a inserção do símbolo “ha” e, como consequência, ocorre a atualização dos percentuais dos símbolos que competem entre si para designá-la.

Por fim, no exemplo da Figura 20, o símbolo “xuma” foi empregado para designar a propriedade “azul”. Sendo assim, na árvore de símbolos ocorre o reforço da palavra “xuma” designar a categoria azul e, por consequência, a probabilidade dela designar a categoria “quadrado” diminui. Paralelamente, na tabela de categorias, mais precisamente na terceira linha, reforça-se a mesma informação. A palavra “xuma” passa a ter 50% de probabilidade de ser fundamentada pela categoria azul e a capacidade de inventar diminui sua probabilidade. Já na última linha, onde a palavra “xuma” também é empregada, porém associada à categoria “quadrado”, ocorre uma penalização. Neste caso, a capacidade de inventar um novo símbolo passa a ser preferencial. Na árvore não há nenhuma penalização, uma vez que a categoria “azul” está associada a um único símbolo.

Portanto, ambas as estratégias são empregadas para que, à medida em que novos símbolos e categorias sejam aprendidos, os percentuais de associação entre (símbolo, categorias) e (categoria, símbolos) sejam ajustados. Vale ressaltar que uma população de

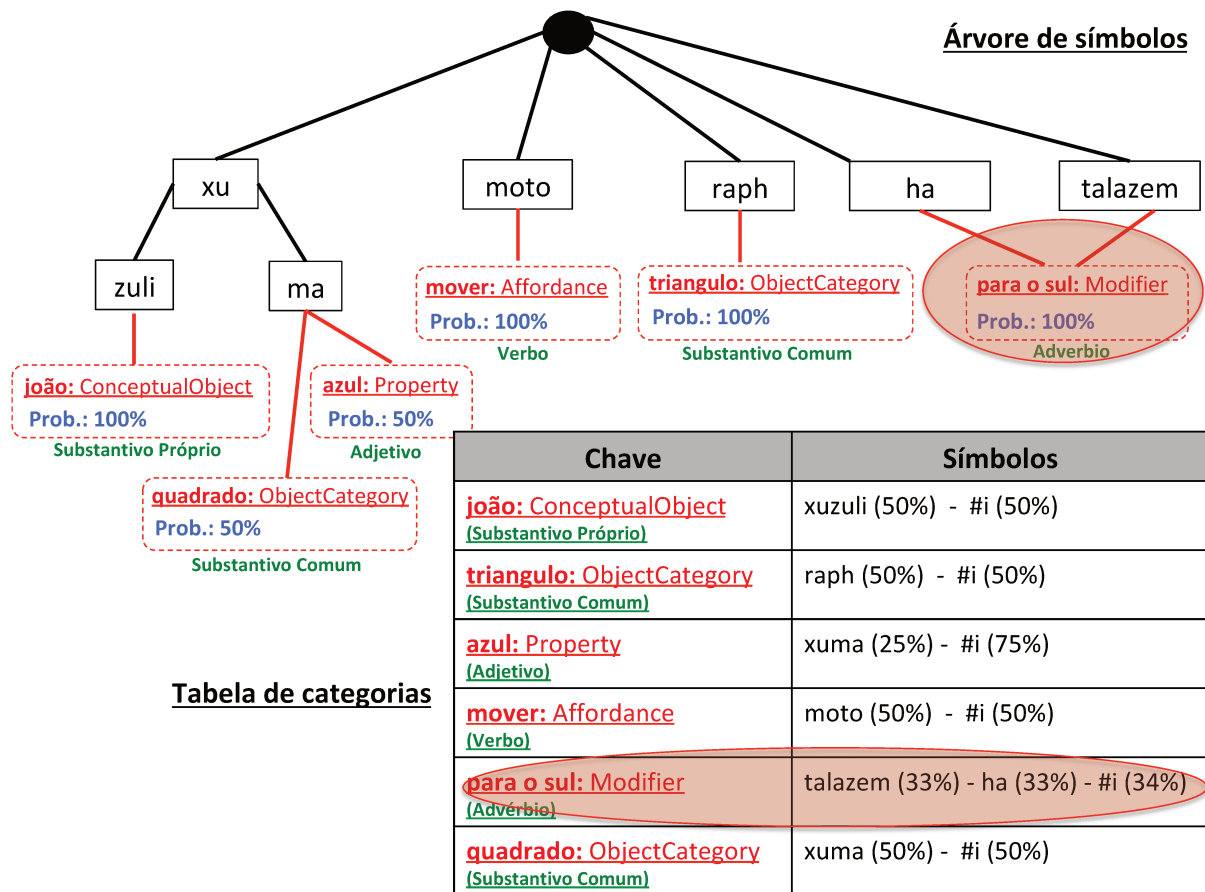


Figura 19 – Exemplo 2 - Ajuste do percentual de associação entre símbolo + categorias na árvore de símbolos e entre categoria + símbolos na tabela de categorias.

falantes, que usa tal mecanismo, estará 100% alinhada gramaticalmente quando empregarem apenas uma (e a mesma) categoria para fundamentar o significado de um símbolo e utilizarem um (e o mesmo) símbolo para designar uma categoria. O mecanismo de aprendizado garantirá que a interoperabilidade semântica ocorra, evoluindo naturalmente ao longo das interações linguísticas estabelecidas.

6.2 Como as Regras Sintáticas são Aprendidas para Definir a Estrutura das Sentenças?

Como visto na seção 6.1, os símbolos que formam uma sentença podem pertencer a diferentes classes gramaticais, quando analisados isoladamente. Porém, eles também podem assumir diferentes funções sintáticas em razão das associações estabelecidas entre si. Embora em uma língua natural possam existir diferentes funções sintáticas, para o escopo deste trabalho, assumiremos somente as seguintes: **sujeito**, **verbo**, **objeto**, **adjunto adnominal** e **adjunto adverbial**.

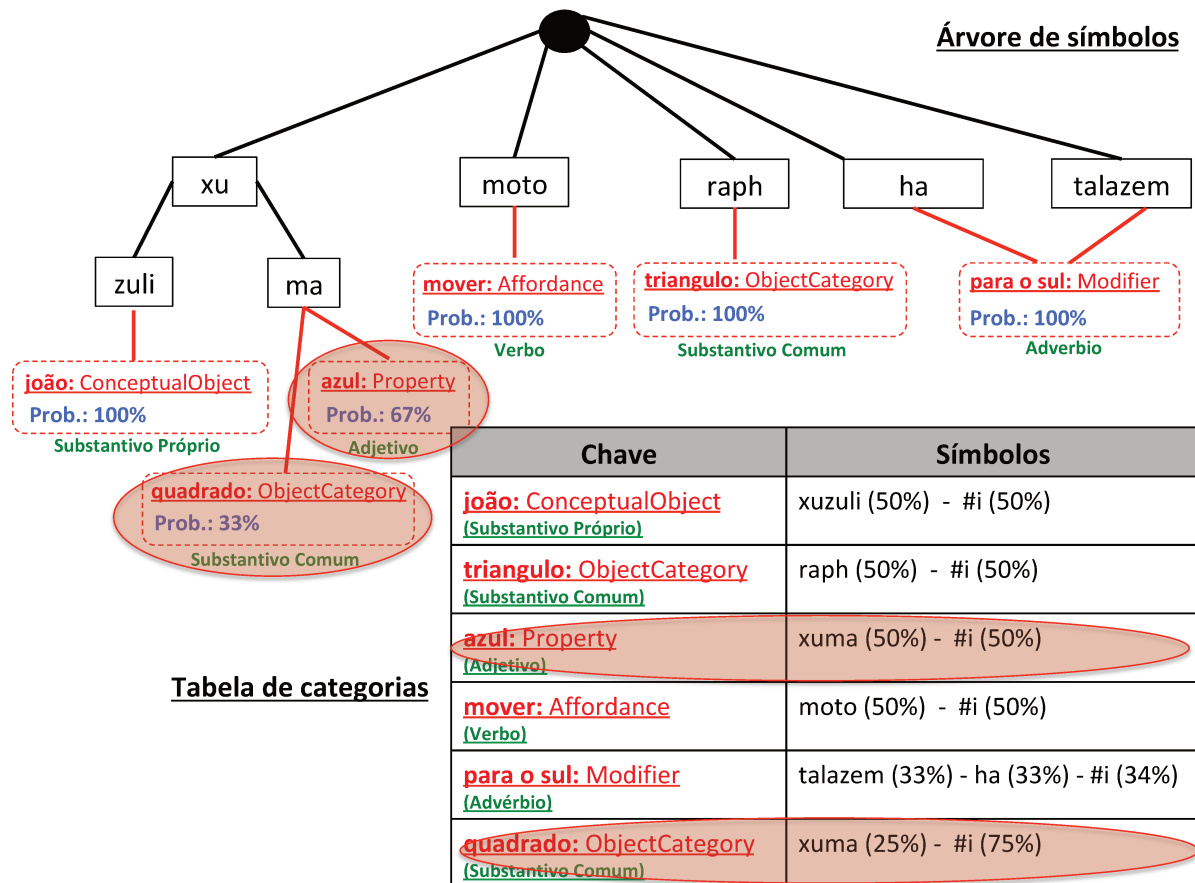


Figura 20 – Exemplo 3 - Ajuste do percentual de associação entre símbolo + categorias na árvore de símbolos e entre categoria + símbolos na tabela de categorias.

O sujeito é o agente responsável por realizar uma ação, sendo representado por um substantivo próprio ou comum. Já os verbos manipulados nas sentenças serão principalmente do tipo transitivo direto, indicando, assim, que necessitam de um objeto para completar o seu sentido. O objeto responderá principalmente às perguntas o “quê?” e “quem?”, indicando o elemento que sofre a ação verbal. Sendo assim, ele também será representado por um substantivo próprio ou comum. O adjunto adnominal e o adjunto adverbial atuam como termos acessórios da sentença, pois funcionam como complementos da mesma e são dispensáveis para a compreensão do enunciado. O adjunto adnominal é o termo que tem valor de adjetivo, servindo para especificar ou delimitar o significado de um substantivo em qualquer que seja a função sintática exercida por este. Já o adjunto adverbial é o termo da sentença que modifica, neste caso específico, o verbo.

Cada papel sintático citado poderia, em princípio, ser correlacionado de qualquer maneira para formar uma sentença. Entretanto, para uma língua específica, essa relação só fará sentido se seguir as regras específicas dessa língua para determinar suas posições. Para o presente trabalho, por uma questão de simplificação, convencionamos que as possibilidades de combinações seriam aquelas apresentadas na Figura 21:

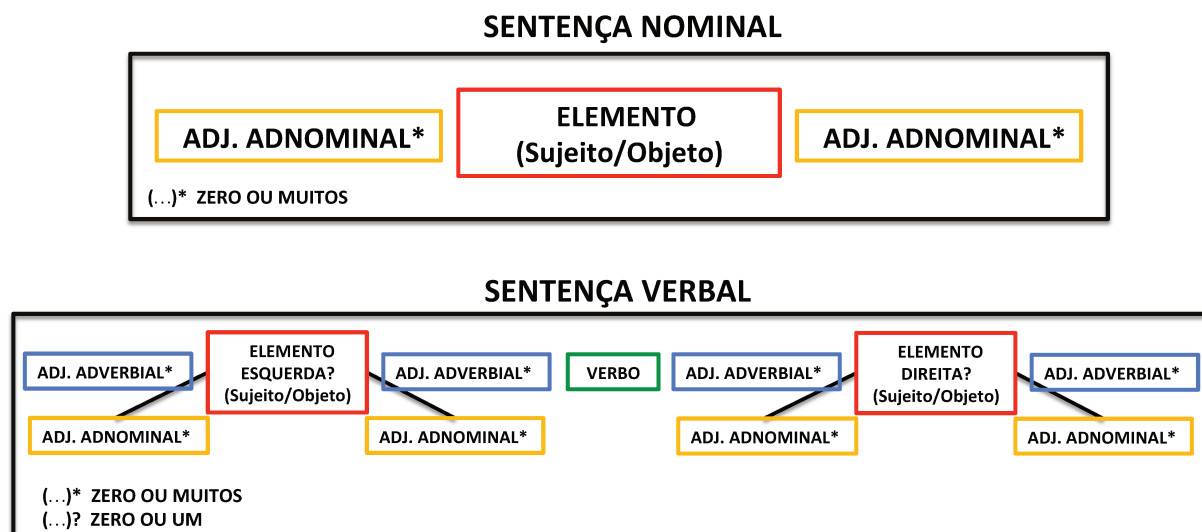


Figura 21 – Estrutura das sentenças nominal e verbal previstas no pacote de linguagem.

Como pode ser observado, há a possibilidade de se formar sentenças nominais e verbais, que obedecerão às seguintes regras:

1. No caso da **sentença nominal**, ela tem, como componente central, pelo menos um elemento, que pode ser o sujeito ou o objeto;
2. O elemento da sentença nominal pode ser qualificado por um ou vários adjuntos adnominais;
3. O adjunto adnominal, quando ocorrer, pode ocupar duas posições: à direita ou à esquerda do elemento. Porém, a definição da sua posição estará atrelada ao domínio a que pertence. Essa decisão foi adotada para poder simular características de línguas naturais, como: em português, adjuntos adnominais que definem cor ficam à esquerda do elemento. Dessa forma, é possível empregar “quadrado amarelo” e não “amarelo quadrado”. Já no caso daqueles que definem tamanho, podem estar tanto à direita quanto à esquerda do elemento. Então, tanto faz usar “quadrado grande ou grande quadrado”;
4. A **sentença verbal** tem como componente central o verbo, que fará a separação entre os elementos sujeito e objeto da sentença;
5. Se o sujeito estiver à esquerda do verbo, o objeto estará à direita ou vice e versa;
6. A ação que está sendo descrita na sentença verbal pode (ou não) ser qualificada por um ou vários adjuntos adverbiais;
7. O adjunto adverbial, quando ocorrer, pode ocupar 4 posições possíveis: à esquerda ou à direita do elemento que está antes do verbo ou à esquerda ou à direita do

elemento que está após o verbo. Porém, a definição da sua posição também estará atrelada ao domínio a que pertence. Dessa maneira podemos simular outras regras de línguas naturais. Por exemplo: em português, é possível empregar o adjunto adnominal “lentamente” em qualquer uma das 4 posições: “lentamente eu pego o livro”, “eu lentamente pego o livro”, “eu pego lentamente o livro” ou ainda “eu pego o livro lentamente”. Já em inglês, eu posso dizer “*slowly, I pick up the book*”, “*I slowly pick up the book*” ou “*I pick up the book slowly*”. Já na terceira posição não se coloca o advérbio (“*I pick up slowly the book*”), pois fica estranho;

8. Por fim, vale ressaltar que uma sentença nominal pode compor uma sentença verbal. Além disso, uma sentença verbal completa tem no mínimo 11 posições válidas, enquanto uma sentença nominal terá 3 posições.

A partir da definição dessas regras, pode existir uma infinidade de combinações de elementos em uma sentença que são consideradas estruturas válidas. Entretanto, algumas dessas possibilidades devem ser aprendidas (e acordadas) durante as interações linguísticas para que possam ser empregadas pelos falantes da língua que está emergindo. Assim, dizemos que uma população de agentes está “sintaticamente alinhada” quando os falantes da população concordarem em utilizar uma mesma regra sintática ou um mesmo conjunto destas. Essa evolução linguística dependerá de um mecanismo de aprendizado, sobre o qual daremos mais detalhes na próxima seção. Vale ressaltar que a possibilidade de recursão sintática não foi pensada neste instante do trabalho.

6.2.1 Sobre o Aprendizado Sintático

A ideia que permeia o aprendizado sintático é fazer com que uma combinação válida dos elementos da sentença seja reforçada e as demais inibidas, de modo que as regras que regem a formação de sentenças possam emergir. Assim, foram disponibilizados alguns métodos de aprendizado por reforço na classe **Language**.

Um desses métodos é o **registerEvent()**. Ele tem como função identificar os elementos sujeito e objeto e registrar suas posições na sentença com relação ao verbo. Outro método é o **registerProperty()**, que irá identificar quais elementos atuam como adjuntos adnominais e, assim, registrar suas posições na sentença, respeitando o domínio a que pertencem. Como dito anteriormente, observar o domínio a que um elemento pertence influenciará no seu significado. Por fim, tem-se o método **registerModifier()**. Este método, atua de maneira semelhante ao anterior, porém se responsabilizando pelos adjuntos adverbiais. Todos esses métodos são empregados para que a estratégia de reforço sinalize a ordem de formação das sentenças, ocorrendo a aprendizagem sintática. Para tal, a seguinte estratégia foi implementada.

- Cada linguagem possui um contador c_a que registra quantas sentenças já tiveram sua ordenação de agente e paciente reforçadas. Inicialmente, c_a tem valor 0.
- Cada domínio possui um contador c_p que registra quantas vezes a ordem de suas propriedades com relação à categoria de objeto já foi reforçada. Inicialmente, c_p tem valor 0.
- Cada domínio também possui um contador c_m que registra quantas vezes a posição de seus modificadores na frase já foi reforçada. Inicialmente, c_m tem valor 0.
- Chamamos de o_r a probabilidade alvo do reforço, isto é, o reforço da posição desejada, que será de 100%.

Estratégia de reforço de ordenação:

1. Atualizamos o valor da ordenação atual o_{n-1} com o valor reforçado o_r de acordo com a seguinte fórmula: $o_n = co_{n-1} + o_r/c + 1$, sendo que c será:
 - c_a , caso estejamos reforçando a ordem do agente e do paciente,
 - c_p , caso estejamos reforçando a ordem das propriedades e
 - c_m , caso estejamos reforçando a posição dos modificadores;
2. O contador é incrementado.

É importante destacar que a estratégia, da forma como foi implementado, vai reforçar imediatamente a ordem imposta para a sentença recebida. No entanto, para que haja uma modificação na probabilidade anterior, uma outra sentença, com outra ordem de seus elementos, deve ser recebida. Além do mais, se o objeto à esquerda for o sujeito, automaticamente o elemento da direita será um objeto. Da mesma forma, se um adjunto adnominal permanecer à direita de um elemento, tal ordem será refletida antes e depois do verbo. Já no caso do adjunto adverbial, a sua posição poderá alternar entre as quatro posições previstas na sentença. Para ficar mais clara a estratégia apresentada, bem como as afirmações feitas, na Figura 22 apresentamos um exemplo.

No tempo T0, é possível observar a ordem inicial e a probabilidade dos elementos constituintes de uma sentença verbal. Para o caso dos adjuntos adnominais e adverbiais, suas posições dependerão do domínio a que estão atrelados. Por esse motivo há a indicação destes na figura também. No exemplo ilustrado, o adjunto adnominal associado ao domínio da cor tem 50% de probabilidade de ficar à esquerda ou à direita do elemento. Já no caso dos adjuntos adverbiais, há duas representações: um adjunto adverbial pertencente ao domínio do tempo e outro pertencente ao domínio da posição. Para cada um deles há 25% de chance de ocuparem as 4 posições destinadas para essa

SENTENÇA VERBAL

Início – T0

Adj Adverbial	Adj Adnominal	Ele Esq	Adj Adnominal	Adj Adverbial	Verbo	Adj Adverbial	Adj Adnominal	Ele Dir	Adj Adnominal	Adj Adverbial
Tempo: 25%	Cor: 50%	Suj: 50% Obj: 50%	Cor: 50%	Tempo: 25%	-	Tempo: 25%	Cor: 50%	Suj: 50% Obj: 50%	Cor: 50%	Tempo: 25%
Posição: 25%		-		Posição: 25%	-	Posição: 25%		-		Posição: 25%

T1

<lentamente> <mover> <triângulo> <para esquerda>

Adj Adverbial	Adj Adnominal	Ele Esq	Adj Adnominal	Adj Adverbial	Verbo	Adj Adverbial	Adj Adnominal	Ele Dir	Adj Adnominal	Adj Adverbial
Tempo: 50%	Cor: 50%	Suj: 100% Obj: 0%	Cor: 50%	Tempo: 50%	-	Tempo: 0%	Cor: 50%	Suj: 0% Obj: 100%	Cor: 50%	Tempo: 0%
Posição: 0%		-		Posição: 0%	-	Posição: 0%		-		Posição: 100%

T2

<João> <moveu> <para esquerda> <triângulo> <amarelo> <rapidamente>

Adj Adverbial	Adj Adnominal	Ele Esq	Adj Adnominal	Adj Adverbial	Verbo	Adj Adverbial	Adj Adnominal	Ele Dir	Adj Adnominal	Adj Adverbial
Tempo: 25%	Cor: 0%	Suj: 100% Obj: 0%	Cor: 100%	Tempo: 25%	-	Tempo: 0%	Cor: 0%	Suj: 0% Obj: 100%	Cor: 100%	Tempo: 50%
Posição: 0%		-		Posição: 0%	-	Posição: 50%		-		Posição: 50%

...

Figura 22 – Emergência das regras sintáticas de uma sentença verbal.

função sintática. Já no caso dos elementos da esquerda e da direita, há 50% de chance dele ser o sujeito e 50% dele ser o objeto ou vice versa.

No tempo T1, ao apresentar a sentença “lentamente mover triângulo para esquerda”, os métodos citados anteriormente identificaram:

- Dois adjuntos adverbiais (“lentamente” e “para esquerda”), sendo o primeiro pertencente ao domínio do tempo (podendo estar na 1a. ou na 2a. posição destinadas aos advérbios) e o segundo pertencente ao domínio de posição (e apresentado na 4a. posição);
- Um objeto (“triângulo”), apresentado como elemento da direita.
- E nenhum sujeito.

Repare que as probabilidades das posições indicadas por um círculo vermelho claro foram reforçadas, enquanto as demais, em negrito, foram inibidas. Como a sentença apresentada não possui sujeito, não se sabe se o adjunto adverbial “lentamente” (pertencente ao domínio do tempo) deverá ficar na 1a. ou na 2a. posição. Sendo assim, as probabilidades de ambas as posições foram reforçadas, ficando 50% para cada uma delas.

Já no caso do adjunto adverbial “para a esquerda” (atrelado ao domínio da posição), que foi apresentado na 4a. posição, foi reforçado com 100%, enquanto o percentual atrelado às demais posições foi inibido. Como o elemento da direita foi interpretado como um objeto, o elemento da esquerda foi reforçado como um sujeito. Portanto, neste exemplo, foi possível perceber como a estratégia de reforço tende a reforçar diretamente (com 100% de probabilidade) a ordem imposta. Vejamos a seguir como ficarão as probabilidades quando uma sentença diferente for recebida.

No tempo T2, a sentença “João moveu para esquerda triângulo amarelo rapidamente” foi apresentada. Neste exemplo, foram identificados:

- Um sujeito (“João”), apresentado como elemento à esquerda do verbo;
- Novamente, dois adjuntos adverbiais (“para esquerda” e “rapidamente”), sendo o primeiro pertencente ao domínio da posição (e apresentado na 3a. posição destinada a essa função sintática) e o segundo pertencente ao domínio do tempo (e apresentado na 4a. posição);
- E um objeto (“triângulo”), apresentado como elemento à direita do verbo.

Neste caso, também houve o ajuste das probabilidades das posições identificadas. Como o sujeito apareceu à esquerda do verbo e o objeto à sua direita (mesma situação da sentença anterior), não houve alteração dos percentuais anteriores, cada um deles permanecendo com 100% de probabilidade. Neste exemplo apareceu, após o elemento da esquerda, o adjunto adnominal “amarelo” (associado ao domínio da cor). Sendo assim, a probabilidade desta posição indicar este adjunto adnominal é de 100% e é refletida nas posições associadas ao elemento da direita, pois trata-se da mesma informação. No caso do adjunto adverbial “para a esquerda” (associado ao domínio de posição), este apareceu na 3a. posição. Sendo assim, o que antes era 100% de probabilidade de ocorrer na 4a. posição, agora ficará dividido, sendo 50% de probabilidade de estar na 3a. e 50% na 4a. posição. Já o adjunto adverbial “rapidamente” (também atrelado ao domínio do tempo) apareceu na 4a. posição. Dessa forma, 1a. e 2a. posições que anteriormente haviam sido reforçadas, agora sofreram uma penalização, ficando cada uma com 25% de probabilidade de demarcarem a posição deste elemento. Já a 4a. posição será reforçada com 50% de probabilidade, passando a ser também uma possível posição para a permanência dos adjuntos adverbiais de tempo.

A estratégia apresentada continua sua atuação, à medida que novas sentenças vão sendo apresentadas. Por conseguinte, as probabilidades de ocupação de cada posição disponível serão reajustadas, até que apenas uma combinação válida emergja e seja empregada de maneira perene na formação das sentenças. Por questões didáticas, no exemplo

apresentado, não houve impedimento de ocupação de nenhuma posição indicada, embora tal restrição possa ocorrer na implementação realizada.

6.3 A Geração e Interpretação Simbólica

Os símbolos manipulados pelo mecanismo de linguagem podem ser de três diferentes naturezas:

- Símbolos isolados, representando as classes gramaticais isoladamente.
- Sentenças nominais, não possuindo um verbo, mas apenas outras classes sintáticas;
- Sentenças verbais, tendo como elemento central o verbo, juntamente com outras classes sintáticas;

Tanto a geração simbólica quanto a interpretação atuarão através do mecanismo de simulação mental, sendo que no caso da geração, o mecanismo provê as categorias que serão nomeadas e no caso da interpretação, restaura as informações brutas que deram origem às categorias e que exercem a fundamentação dos símbolos em questão. Dessa forma, há um intercâmbio entre os pacotes *Perception*, *Ontology* e *Language*. Vale ressaltar que as categorias ontológicas recuperadas a partir do evento ou isoladamente estão associadas às informações brutas armazenadas nas estruturas de memória, mantidas pelo pacote de percepção. Portanto, faz-se necessário explicar essa correlação:

- O que no mecanismo de linguagem configura-se como um **símbolo isolado**, no modelo de representação pode ser uma **categoria de objeto**, um **objeto conceitual**, uma **propriedade** ou ainda uma *affordance* apenas.
- O que no mecanismo de linguagem é configurado como uma **sentença nominal**, no modelo de representação poderá ser uma **categoria de objeto** ou um **objeto conceitual** juntamente com pelo menos uma **propriedade**.
- O que no pacote de linguagem é configurado como uma **sentença verbal**, no pacote de categorias ontológicas é um **evento**, disparado por meio da aplicação de uma *affordance*.

A seguir apresentamos como os algoritmos de geração e interpretação simbólica funcionam.

6.3.1 Algoritmo para Obtenção de Símbolos

O algoritmo para obtenção de símbolos tem como objetivo expressar linguisticamente instâncias de categorias ontológicas criadas pelo mecanismo de simulação mental, quando atuando no modo percepção. Ele usa o método **elementToSymbol()**, que tem como base a tabela de categorias, que mapeia categorias em símbolos. Seu funcionamento pode ser resumido pelos seguintes passos descritos a seguir:

1. Ao receber uma instância de categoria ontológica, utilizar a tabela de categorias para recuperar os símbolos atrelados a ela.
 - a) Se a categoria for um evento, recuperar, separadamente, os símbolos de cada instância de categoria que o compõe.
2. Se para uma categoria houver mais de um símbolo para expressá-la, realiza-se um sorteio ponderado entre eles, com base nos percentuais registrados. O símbolo vencedor deve ser utilizado para expressar a instância de categoria em questão naquele instante.
 - a) Se a categoria for única (isto é, se ela for uma categoria de objeto ou um objeto conceitual - discriminado - ou uma *affordance* ou uma propriedade) recupera-se um símbolo isolado para expressá-la;
 - b) Caso contrário, uma sentença é formada:
 - i. Se a categoria for um evento, uma sentença verbal deve ser formada.
 - ii. Se a categoria for um objeto conceitual (que ainda não foi discriminado), uma sentença nominal deve ser formada. A sentença deve conter a categoria do objeto e as propriedades necessárias para discriminá-lo dos objetos semelhantes.
 - iii. Os símbolos devem ser ordenados de acordo com as regras sintáticas registradas.
3. O símbolo de alta-ordem obtido pode ser expresso.

6.3.2 Algoritmo para Interpretação de Símbolos

O algoritmo de interpretação simbólica tem como objetivo receber um símbolo de alta-ordem e gerar uma simulação mental que corresponda à interpretação desse símbolo. Essa simulação mental é uma instância de suas respectivas categorias ontológicas e informações brutas, consideradas como o fundamento de seu significado. Para

isso é utilizado o método **symbolsToElement()**, que tem como base a árvore de símbolos (estrutura de dados que indica as categorias associadas a cada símbolo). O seu funcionamento pode ser resumido pelos seguintes passos:

1. Receber os símbolos em questão.
2. Percorrer a árvore de símbolos para recuperar as categorias atreladas a cada símbolo.
 - a) A cada transformação detectada, remover a transformação do símbolo e repetir o passo 1.
3. Caso seja uma sentença, um filtro será aplicado para identificar as combinações de categorias que respeitem as regras apresentadas na seção 6.2. Esse filtro funciona da seguinte maneira: ao receber o símbolo composicional “xuzuli moto raph” realiza-se uma busca na árvore de símbolos em busca das categorias que fazem a fundamentação de cada símbolo isoladamente. Para o símbolo “xuzuli” temos as categorias “João:ConceptualObject” e “triângulo:ObjectCategory”, para o símbolo “moto” temos as categorias “mover:Affordance” e “azul:Property” e para o símbolo “raph” temos as categorias “Maria:ConceptualObject” e “quadrado:ObjectCategory”. De acordo com as categorias recuperadas e com o que está registrado na tabela das regras sintáticas, pode-se concluir que as regras que permitem combinar de maneira válida essas categorias são as seguintes:
 - RS1: (Sujeito)(Verbo)(Objeto) e
 - RS2:(Objeto)(verbo)(Sujeito)
 - a) Para cada combinação é associada uma probabilidade obtida a partir das regras gramaticais e sintáticas. Por exemplo, os símbolos “xama gili motolo” podem ser interpretados como: “João (70%) mover (10%) quadrado (30%)”, gerando um grau de confiança de 1.1 (70% + 10% + 30%). Uma outra alternativa seria “Maria(30%) pintar (90%) círculo (70%)”, gerando um grau de confiança de 1.9 (30% + 90% + 70%). Adicionando ao cálculo a probabilidade da regra sintática <Sujeito><Verbo><Objeto> que é 80%, isso faria com que o grau de confiança da primeira interpretação subisse para 1.9 e o da segunda interpretação para 2.7. Nesse exemplo, “Maria pinta círculo” tem mais chances de ser escolhido como interpretação dos símbolos “xama gili motolo”. Portanto, esse raciocínio é usado para calcular a probabilidade de cada interpretação gerada.
4. Um sorteio é realizado para escolha da interpretação, sendo ponderado pela probabilidade associada a cada interpretação.
5. A quantidade de elementos de cada tipo é contada para definir qual categoria se formou:

- a) Se Propriedade + (Objeto Conceitual ou Categoria de Objeto): formou-se uma sentença Nominal. Portanto, o resultado será um **Objeto Conceitual**;
 - b) Se *Affordance* + (Categoria de Objeto ou Objeto Conceitual ou Modificadores ou transformação temporal): formou-se uma sentença verbal. Portanto, o resultado será um **Evento**;
 - c) Se somente uma *Affordance*, o resultado será uma **Affordance**;
 - d) Se somente uma Propriedade, o resultado será uma **Propriedade**;
 - e) Se somente uma Categoria de Objeto, o resultado será uma **Categoria de Objeto**;
 - f) Se somente um Objeto Conceitual, o resultado será um **Objeto Conceitual**;
6. Se existirem objetos conceituais entre as categorias interpretadas, utilizar as estruturas de memória para associar os objetos interpretados a objetos já conhecidos para configurar a simulação mental de interpretação das categorias.
7. Retornar as categorias interpretadas.

6.4 Sobre as Transformações

As transformações têm como objetivo modificar a estrutura de um símbolo, sem fazer com que ele altere sua classe gramatical. Na implementação proposta, há a possibilidade de trabalhar com transformações de variados tipos, como tempo, número, gênero e tamanho.

No caso da transformação de tempo, ela será aplicada aos verbos, no contexto de uma sentença e as demais transformações serão aplicadas a substantivos e adjetivos. As transformações de tempo permitidas são presente, passado e futuro. Já a transformação de número é somente o plural. No caso da transformação de gênero, há o masculino e o feminino e na transformação de tamanho, o aumentativo e o diminutivo. As transformações poderão ser aplicadas tanto na geração quanto na interpretação simbólica, mas por questões de teste dessa funcionalidade optamos, para este momento, por implementar somente a transformação de tempo. A seguir vejamos mais detalhes de como ela foi implementada.

6.4.1 A Transformação de Tempo

A transformação temporal pode ser aplicada de maneira simples e direta. Como dito anteriormente, ela se aplica à *affordance* de um evento. Para isso, o evento possui, em sua estrutura, a possibilidade de se associar um tempo verbal, localizando o

evento no passado ou no futuro. Durante a geração de símbolos, há a possibilidade de se escolher um tempo verbal para o evento. Feita a escolha, o símbolo que representa a *affordance* será localizado e sobre ele será aplicada a transformação correspondente.

Vale ressaltar que durante a interpretação simbólica, as transformações devem ser desfeitas para que seja possível interpretar o símbolo. Para isso, durante a ação do algoritmo de interpretação, ao iterar por cada símbolo da sentença, tenta-se desfazer a transformação que ele recebeu.

6.5 Considerações Finais

Neste capítulo introduzimos o funcionamento do pacote de linguagem, onde propusemos uma extensão do mecanismo de simulação mental para tratar a questão da linguagem, conectando-a ao processo de simulação mental - modo percepção. Ao longo do texto clarificamos o significado do termo “símbolo de alta-ordem”, empregado para designar tanto símbolos isolados quanto sentenças, mais especificamente sentenças verbais e nominais.

Também apresentamos as principais classes que compõem o modelo, bem como os métodos que implementam o mecanismo de aprendizado associativo, necessário para a evolução de uma gramática de escopo simplificado.

Por uma questão didática, apresentamos primeiro as estratégias para evolução do aprendizado gramatical, i.e., as estratégias que promovem a associação entre símbolos isolados e as categorias ontológicas empregadas em sua fundamentação. Em seguida, apresentamos as estratégias para a evolução do aprendizado sintático, i.e., aquelas que dizem respeito às regras que regem a estruturação das sentenças empregadas para designar eventos e episódios.

Por fim, apresentamos o funcionamento do mecanismo de simulação mental no modo de linguagem, ou seja, os processos de geração e interpretação de símbolos. Além dessas funcionalidades, mostramos também como alguns símbolos podem sofrer eventuais transformações temporais.

As funcionalidades apresentadas aqui de modo genérico serão aplicadas, no próximo capítulo, em um jogo de linguagem, a fim de validar a funcionalidade dos mecanismos propostos. Nosso intuito é demonstrar que, de posse das estruturas de representação, o mecanismo é capaz de evoluir um sistema de linguagem comum, para uma população de agentes, garantindo assim a interoperabilidade semântica entre eles. Os detalhes sobre esse jogo estão no capítulo 7.

7. EXPERIMENTOS COMPUTACIONAIS

Neste capítulo, apresentamos os experimentos executados para validar o funcionamento do *framework* computacional implementado para o mecanismo de simulação mental proposto. Como visto no Capítulo 5, a estrutura do *framework* é formada por processos intermediários (modelar informações sensoreadas; descobrir categorias ontológicas; perceber eventos; nomear categorias ontológicas e interpretar símbolos composicionais), que em conjunto conferem ao mecanismo de simulação dois modos de atuação distinto - o modo percepção e o modo linguagem. Dessa forma, os experimentos testaram a atuação dos processos que julgamos ser os mais relevantes para funcionamento do mecanismo, bem como o correlacionamento entre eles para prover os modos de atuação: percepção e linguagem.

Sendo assim, foram executados dois tipos de experimentos: o primeiro teve como objetivo avaliar um algoritmo bio-inspirado que foi implementado para atuar no processo de descobrimento de categorias ontológicas.

Já o segundo teve como objetivo avaliar o funcionamento do mecanismo de simulação mental atuando nos modos percepção e linguagem. Nesse experimento, exercitamos os processos de percepção de eventos, nomeação e interpretação de símbolos composicionais. As categorias usadas para a categorização do processo de percepção foram configuradas previamente, utilizando o método do experimento 1. Portanto, no experimento 2 não revalidaremos o processo de descobrimento de categorias.

Vale ressaltar que em nenhum dos experimentos estabelecidos mencionamos a validação do processo de modelagem de informações brutas. Isso aconteceu porque partimos do pressuposto de que esse é um processo intrínseco ao mecanismo, sem o qual nenhum outro funcionaria. Assim, ao garantir que processos mais elaborados ocorram, garantimos também o seu funcionamento.

Os resultados e comentários adicionais sobre cada tipo de experimento serão apresentados nas seções correspondentes.

7.1 Experimento 1: A Segmentação dos Espaços Conceituais

Como vimos na seção 2.5, ao se representar objetos em um espaço conceitual, regiões convexas podem ser formadas e objetos protótipos são eleitos para representá-las. Dessa maneira, conceitos podem ser definidos na forma de protótipos. Com essa proposta, Gärdenfors argumenta que a teoria dos espaços conceituais pode funcionar como um *framework* semântico e, assim, auxiliar na fundamentação simbólica. Embora

sua teoria tenha *insights* promissores, a questão que permanece é como implementá-la computacionalmente e empregá-la em situações que envolvam linguagem e comunicação. O experimento apresentado nesta seção tem como objetivo apresentar um ferramental computacional bio-inspirado que seja capaz de criar um conjunto de objetos prototípicos para que diferentes tipos de categorias ontológicas possam ser reificadas.

Para verificar se as categorias descobertas são passíveis de serem empregadas na percepção de objetos e eventos e, conseqüentemente aplicados na nomeação e interpretação simbólica, um teste adicional foi executado através da implementação de um tipo de jogo de linguagem. O jogo de linguagem se mostrou ideal para este contexto, pois para garantir que atos linguísticos ocorram entre os falantes de uma língua, esses devem categorizar os objetos que compõem um ambiente e indicá-los, expressando símbolos linguísticos. Para uma correta compreensão do que foi dito, eles devem interpretar esses símbolos.

Entretanto, como o objetivo principal deste experimento foi exercitar a capacidade do algoritmo escolhido na descoberta de categorias ontológicas, algumas simplificações foram feitas na estrutura do *framework* e, conseqüentemente, no funcionamento dos demais processos que o compõem. Assim sendo, o processo de percepção vai detectar apenas as propriedades dos objetos. O processo de nomeação criará apenas símbolos isolados para descrever objetos por meio de suas qualidades. Em razão dessa simplificação, utilizamos como estrutura de dados apenas a tabela de categorias. Além do mais, os objetos são estáticos no ambiente. Dessa forma, o algoritmo de rastreamento não foi aplicado em sua totalidade (não se aplicou a detecção de *affordances*, eventos, ou episódios).

7.1.1 Espaços Conceituais e Mapas Auto-Organizáveis Evolutivos

Na área de aprendizado de máquina, os algoritmos de *clustering* são métodos capazes de detectar semelhanças, regularidades e correlações nos dados de entrada, particionando-os em grupos (*clusters*) (SILVA *et al.*, 2010). Espaços conceituais, como mostrado por Gärdenfors (2004) atuam de maneira semelhante, onde as regiões convexas discriminadas pelos domínios são usadas para categorizar objetos que compartilham propriedades semelhantes. Portanto, um espaço conceitual funciona da mesma maneira que um algoritmo de clusterização.

Existem vários algoritmos de clusterização disponíveis na literatura (Jain *et al.* (1999), Xu *et al.* (2005)) que poderiam ser usados para detectar os domínios de um espaço conceitual. Nossa escolha baseou-se na compatibilidade com as definições da teoria de espaços conceituais, buscando a fundamentação de símbolos linguísticos. Para essa escolha, definimos os seguintes critérios:

1. O algoritmo escolhido deveria ter plausibilidade biológica.
2. Deveria ter a capacidade de aprender novos conceitos durante as interações, sem treinamento prévio.
3. O número de classes deveria ser desconhecido, *a priori*.

Com base nessas premissas, selecionamos uma rede neural artificial auto-organizável, chamada *Evolving Self-Organizing Map* (ESOM), proposta por Deng e Kasabov (2000). A rede ESOM baseia-se nos mapas auto-organizáveis (SOM - do inglês, Self-Organizing Maps)¹, uma rede neural clássica proposta por Teuvo Kohonen (1982), que foi desenvolvida baseada nos mapas tonotópicos encontrados nos cérebros animais e humanos e relatados na literatura de neuro-ciência. De acordo com Silva *et al.* (2010), a rede SOM foi inspirada no córtex humano e é comumente aplicada à resolução de problemas que envolvem classificação de padrões e agrupamento de dados. Porém, ela tem uma limitação que impede sua utilização na aprendizagem de categorias de um espaço conceitual: a necessidade de se definir a estrutura de seu mapa *a priori*.

O fato do ESOM ser um algoritmo evolutivo torna-o particularmente interessante para o particionamento de espaços conceituais, principalmente devido à sua capacidade de realizar aprendizagem *on-line* sem supervisão. A estrutura de rede formada durante a fase de aprendizagem é auto-adaptativa e incremental. Sendo assim, novos *clusters* e, possivelmente, *outliers* podem aparecer e desaparecer, criando uma topologia de acordo com a organização de dados. Estas características dão mais flexibilidade e eficiência para o ESOM, permitindo que ele crie boas representações para os dados de entrada. Sendo assim, neste trabalho, a configuração dos nós da rede será utilizada para segmentar o espaço conceitual em regiões semânticas (*clusters*/regiões convexas), onde cada região pode representar uma propriedade, identificada por meio de um nó que funcionará como seu protótipo.

Como o objetivo deste experimento é validar a aplicabilidade do ESOM como um ferramental capaz de particionar o espaço conceitual, uma simplificação foi feita na estrutura do processo de representação, de modo que instanciaremos apenas propriedades. A instanciação das demais categorias de objetos ficarão para trabalhos futuros.

Além do mais, aqui não iremos comparar o desempenho do ESOM com outras ferramentas, pois este trabalho já foi feito por Deng e Kasabov (2000). Neste caso, eles compararam o desempenho de uma rede ESOM com outras redes neurais, tais como a SOM e a Neural Gas, e os resultados indicaram que a rede ESOM foi a mais rápida e mais eficaz na resolução dos problemas abordados. Mais detalhes sobre esses testes podem ser encontrados em (DENG; KASABOV, 2000).

¹ A rede SOM também é conhecida como Mapa Auto-Organizável de Kohonen ou, simplesmente, Rede de Kohonen.

7.1.2 A Implementação e Funcionamento da Rede Neural ESOM

Neste trabalho o ESOM foi implementado como uma aplicação Java, seguindo o algoritmo descrito por [Deng e Kasabov \(2000\)](#). A rede inicia sem nenhum nó e, durante cada iteração de aprendizagem, ela se atualiza para categorizar os dados de entrada, criando novos nós e novas conexões, quando necessário (ou seja, quando a rede não possui nó ou a distância euclidiana entre o dado de entrada e o nó mais próximo é maior do que o limite estabelecido ϵ). O parâmetro ϵ é muito importante para o desenvolvimento da rede, uma vez que controla sua taxa de crescimento. No caso do nosso trabalho, ele foi ajustado com base na tarefa e nos dados que estávamos trabalhando.

Quando um novo nó é criado, ele representa exatamente o dado de entrada da iteração corrente. Este nó pode sofrer alterações em seu peso, se houver uma correspondência entre ele e dados de entrada posteriores. Porém, se a distância entre os dados de entrada e o nó mais próximo é menor do que ϵ , as conexões deste nó são atualizadas e nenhum novo nó é adicionado. As conexões entre os nós têm um papel fundamental em manter o relacionamento entre nós vizinhos. Sua força (relacionamento) é determinada pela distância, que é atualizada pela taxa de aprendizagem $0 < \gamma < 1$. A cada t passos, a ligação entre os dois nós mais distantes (ou seja, os nós com a ligação mais fraca) é podada.

A Figura 23 demonstra o funcionamento do algoritmo, ilustrando a evolução de uma rede ESOM em três momentos distintos. Neste exemplo, seis nós foram criados e numerados na ordem de sua criação. Cada novo nó mantém uma relação de proximidade com os dois nós mais próximos. Por exemplo, o nó 6 foi criado e ligado com os nós 3 e 5. Depois de algum tempo, a ligação entre os nós 1 e 3 foi podada. A poda ocorre quando a distância entre dois nós é grande (indicando uma ligação fraca). Cada nó tem uma área delimitada, representada por uma hiper-esfera de raio ϵ .

7.1.3 Validação: Particionando Espaços Conceituais com ESOM

A fim de validar a nossa implementação da rede ESOM e verificar se ela poderia segmentar espaços conceituais, criamos um conjunto de dados sintéticos com 1000 pontos, compostos por 2 dimensões cada e distribuídos aleatoriamente entre 12 classes. Para fazermos essa distribuição, a cada ponto da amostragem foi acrescentado um ruído aleatório seguindo uma distribuição normal ($\mu = 0$, $\sigma = 2$). Os pontos resultantes são mostrados na Figura 24. Já suas classes são representadas por cores diferentes na Figura 25. É fácil perceber que não há uma distinção clara entre classes vizinhas. Essa configuração foi gerada de maneira propositada para verificarmos a capacidade do ESOM de generalizar sua classificação, mesmo na presença de dados ruidosos.

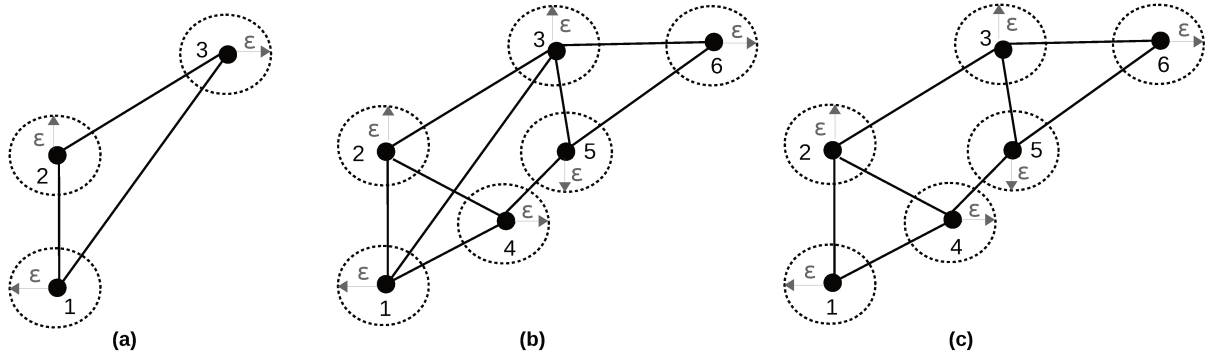


Figura 23 – Rede ESOM em três momentos diferentes. Em (a) os três primeiros nós são criados (nós 1, 2 e 3); Em (b) três novos nós são adicionados (os nós 4, 5 e 6); Em (c) a ligação mais fraca é removida - Adaptado de [Deng e Kasabov \(2000\)](#).

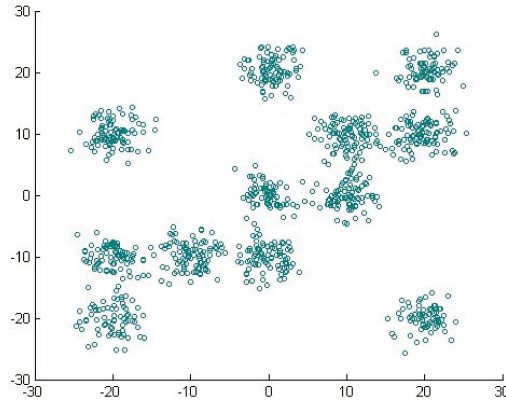


Figura 24 – Representação gráfica de nosso conjunto de dados sintéticos, composto por 12 classes (com 2 dimensões cada) e 1000 pontos de dados. Cada ponto foi gerado aleatoriamente e um ruído foi aplicado a eles, seguindo uma distribuição normal ($\mu = 0$, $\sigma = 2$).

Depois de definir os parâmetros de configuração da rede ESOM (para reprodutibilidade: **threshold** $\epsilon = 9.0$ e **taxa de aprendizagem** $\gamma = 0.4$), a execução da rede começa. A cada iteração, um ponto foi apresentado ao ESOM para que pudesse ser classificado. Desta forma, a representação do conjunto de dados foi aprendida em tempo real, sem qualquer informação prévia. O resultado da primeira classificação feita pelo ESOM, pode ser visto no diagrama de Voronoi apresentado na Figura 26, onde os pontos pretos representam os nós da rede e cada célula do diagrama sua área de representação.

Na Figura 26, os nós da rede ESOM atuam como pontos prototípicos, representando a configuração do conjunto de dados. Cada um deles segmenta o espaço em áreas semânticas (representadas pelas células do Voronoi), com a forma de regiões convexas, e determinando que cada ponto dentro de uma respectiva região compartilha a

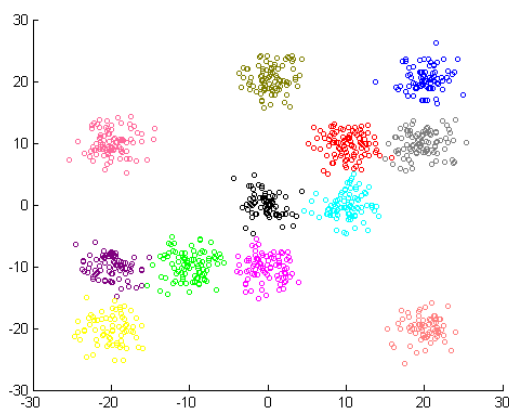


Figura 25 – Representação gráfica de nosso conjunto de dados sintético apresentado na Figura 24, com cada classe original representada por uma cor distinta.

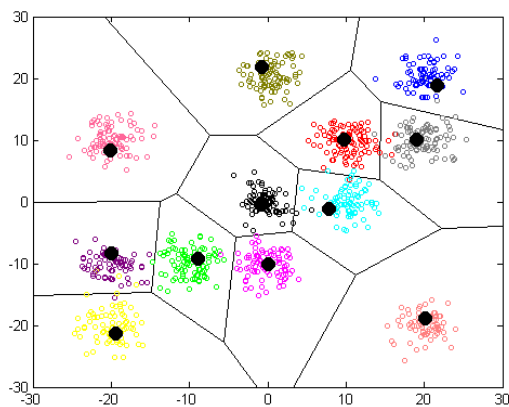


Figura 26 – Pontos rotulados em tempo real através de uma técnica não-supervisionada (ESOM) e dispostos em um diagrama de Voronoi após o final da 1ª fase de classificação. Cada cor representa um rótulo diferente e os pontos pretos grandes representam os protótipos (nós da rede), determinados pelo ESOM.

mesma propriedade, ou seja, pontos pertencentes a uma mesma região são semelhantes. De acordo com [Gärdenfors \(2004\)](#), o diagrama de Voronoi gerado fornece uma resposta geométrica de como medidas de similaridade, juntamente com pontos protótipos, podem determinar um conjunto de categorias.

Ainda pode-se observar na Figura 26 que cada ponto é colorido de acordo com o rótulo atribuído pelos nós do ESOM durante a primeira fase de treinamento. Embora a divisão espacial e os rótulos tenham sido, em sua maioria, corretamente definidos, há ainda alguns pontos isolados que não foram classificados corretamente, levando-nos a concluir que os pontos prototípicos (os nós da rede) estavam se movendo à medida que classificavam outros pontos da amostra. Porém, ao submeter a mesma amostra de dados sucessivas vezes ao ESOM, percebemos que, ao longo do tempo, as regiões semânticas determinadas tendem a se estabilizar. Na Figura 27 é possível observar o desempenho da

rede após 10 iterações sucessivas de classificação da mesma base de dados.

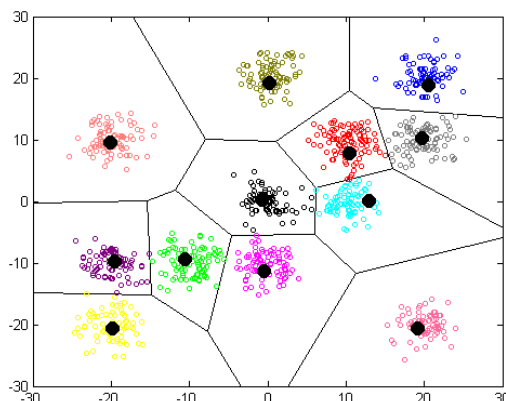


Figura 27 – Pontos rotulados em tempo real pelo ESOM e dispostos em um diagrama de Voronoi após o final da 10ª fase de classificação.

Observando a Figura 27, percebemos que o número de *outliers* diminuiu em relação à Figura 26. Além disso, a diferença entre as regiões convexas formadas pela primeira e a décima iterações também é pequena, indicando que o ESOM criou uma boa representação para o problema abordado, mesmo depois de classificar os dados somente uma vez.

7.1.4 Jogo de Linguagem Implementado

O desempenho da rede ESOM no processo de descobrir categorias foi validado por meio da execução de um jogo de adivinhação. Neste caso, os jogadores, aqui representados por agentes de software, devem representar o seu conhecimento sobre o ambiente em espaços conceituais e utilizar os conceitos para fundamentar os símbolos trocados durante a sua comunicação.

Como apresentado no capítulo 3, em um jogo de linguagem, uma população de agentes deve usar nomes (símbolos) para referenciar tipos específicos de objetos presentes em seu ambiente. A atribuição de nomes é local entre dois agentes específicos, mas os nomes podem se espalhar pela população, se houver interações entre diferentes pares de agentes. No caso do nosso experimento, além do sincronismo linguístico entre os agentes, gostaríamos também que significados compartilhados pudessem emergir para os nomes trocados entre eles (o que chamamos de interoperabilidade semântica).

Durante o jogo, os participantes constroem um modelo interno (categorias ontológicas) usando suas informações sensoriais e suas memórias. Nesse caso, a linguagem age como uma função que os agentes empregam para atingir seus objetivos. Dado seu modelo interno e propósito, o agente falante categoriza uma estrutura semântica, que é

usada pelo sistema linguístico, para produzir um nome. O mesmo acontece com o ouvinte que, ao receber um nome, usa seu modelo interno para encontrar um conceito semântico correspondente para interpretá-lo.

O jogo de adivinhação que implementamos nesta seção é semelhante ao jogo de nomeação implementado por Wellens (2012), porém com um estágio extra para categorização, implementado pelo ESOM e o processo perceptual, no protocolo do jogo. Esta é uma mudança necessária, porque no jogo de Wellens os nomes têm um significado apenas quando o objeto referido está fisicamente presente no contexto. Mostramos que essa mudança não modifica os resultados mostrados por Wellens (2012), mas sim os ampliam, permitindo que outras situações sejam exploradas, como a interpretação de texto (sem a presença de objetos físicos) e a variabilidade de objeto dentro da mesma classe.

O jogo de adivinhação foi desenvolvido com uma população de 50 agentes tendo a mesma configuração perceptual, motora e linguística em um ambiente com objetos descritos por 5 qualidades distintas, cada uma delas descrita por um vetor composto por 4 dimensões de qualidade, com valores brutos aleatórios no intervalo $[0,1]$. Os agentes devem chegar a um consenso a respeito dos símbolos isolados empregados para nomear as qualidades dos objetos.

A cada duelo do jogo, um contexto é gerado com 3 objetos e dois agentes (falante e ouvinte) são escolhidos aleatoriamente da população e confrontados com o mesmo contexto. Em nossa implementação, os agentes extraem as informações brutas dos objetos para segmentar seus espaços conceituais (modelos internos) e criar as propriedades utilizadas na categorização dos símbolos. Em seguida, o seguinte protocolo da Figura 28 é executado.

1. Agentes sensoreiam o contexto e extraem informações brutas dos objetos;
2. Cada agente segmenta seu espaço conceitual e cria as categorias ontológicas.
3. O falante mentalmente escolhe uma propriedade;
4. O falante associa um símbolo de alta ordem (uma palavra única, trissilábica) à categoria e o expressa;
5. O ouvinte interpreta o símbolo, a partir de suas categorias e experiências passadas;
6. O ouvinte tenta indicar a propriedade referente;
7. O falante concorda ou discorda do ouvinte;
8. Ambos os agentes têm a oportunidade de adaptar o seu dicionário léxico (tabela de categorias). No caso de concordância, a associação entre o nome e a categoria é

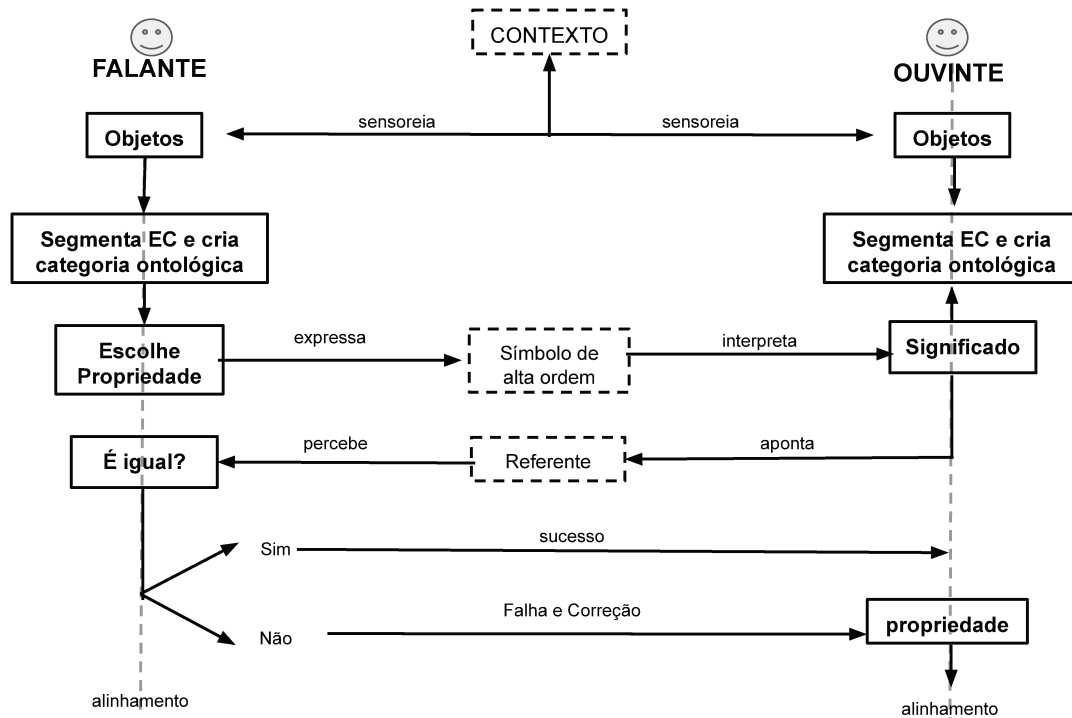


Figura 28 – Protocolo do Jogo de Adivinhação estabelecido. Os quadrados representam os dados produzidos pelos agentes. Os textos refere-se aos processos inerentes a cada fase. Os processos podem falhar. Entretanto, existem estratégias de diagnóstico e reparo associadas a cada um deles. Um jogo de adivinhação consiste em milhares dessas interações.

reforçada. No caso de discordância, a associação é punida e o falante informa qual foi a propriedade mencionada.

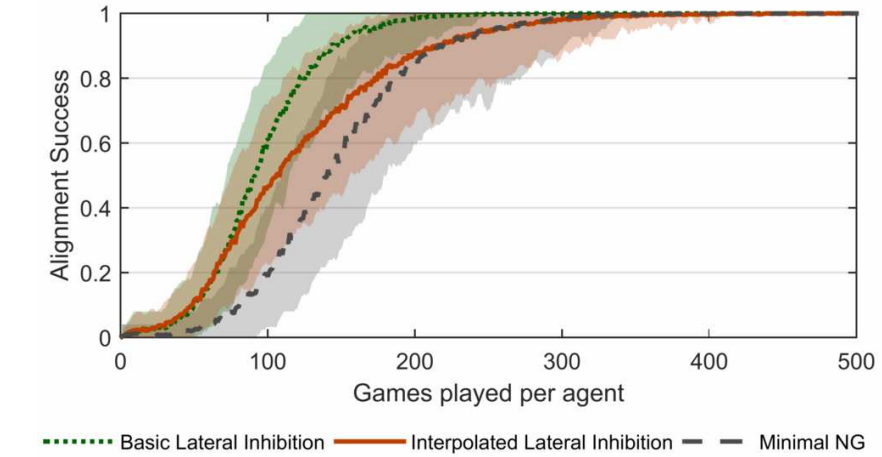
9. Um novo jogo pode ser iniciado.

As estratégias de aprendizado empregadas para reforço e punição dos nomes que competem para descrever uma categoria são aquelas descritas no capítulo 3, na seção 3.3.1 - Jogo de Nomeação Mínimo, Inibição Lateral Básica e Inibição Lateral Interpolada (respectivas siglas em inglês: MNG, BLI e ILI).

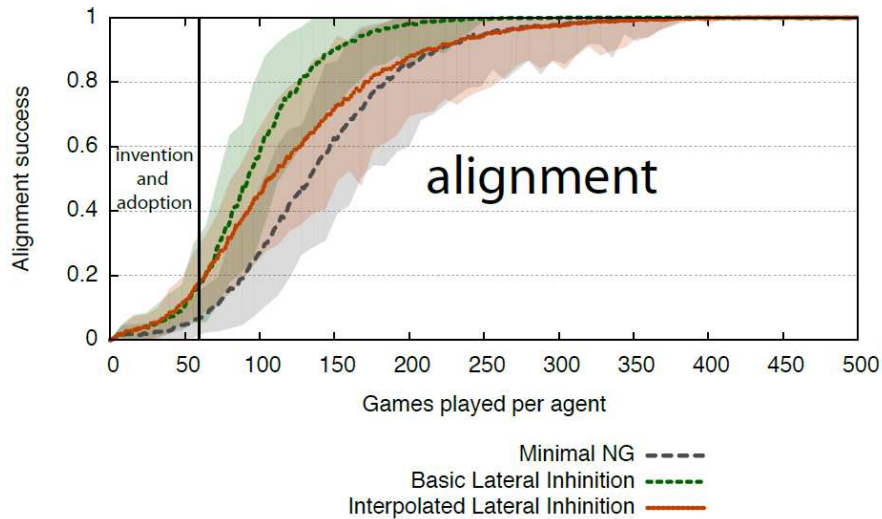
7.1.5 Resultados e Discussões

Para avaliar o desempenho dos agentes no jogo, mensuramos o alinhamento populacional, isto é, o sucesso de alinhamento, que é dado por: (i) o sucesso comunicativo, ou seja, a capacidade do ouvinte de identificar corretamente o nome pronunciado pelo falante, apontando para o referente correto e (ii) a preferência do agente em relação ao nome, ou seja, a medida de quanto o falante e o ouvinte escolhem o mesmo nome para apontar para um objeto. Se ambos os critérios são atendidos, o sucesso do alinhamento é 1, caso contrário, 0.

Realizamos um total de 100 experimentos, cada um deles com 12500 iterações e uma média de 500 duelos por agente². O alinhamento médio da população foi calculado para cada estratégia de nomeação (MNG, BLI e ILI) e os resultados são mostrados na Figura 29. Para questões de comparação, na mesma Figura apresentamos os resultados obtidos por Wellens (2012).



(a) Sucesso de alinhamento alcançado por nossa implementação.



(b) Sucesso de alinhamento alcançado por Wellens (2012). Reproduzido de Wellens (WELLENS, 2012).

Figura 29 – Sucesso de alinhamento para as estratégias de inibição lateral e jogo de nomeação mínimo. A Figura(a) mostra o sucesso de alinhamento alcançado por nosso método e a Figura (b) mostra o sucesso de alinhamento alcançado por Wellens (2012). Ambas as estratégias de inibição lateral usaram como parâmetro: $\delta_{inc} = \delta_{dec} = 0.1$, $\delta_{inh} = 0.5$ and $s_{initial} = 0.5$.

Ambos os resultados, do nosso experimento e de Wellens (2012), indicaram que as estratégias BLI e ILI têm convergência rápida no início do experimento, enquanto

² Para uma média de 500 duelos, temos $(2 * (\text{número de jogos} \div (\text{tamanho da população}))$

a estratégia de MNG converge a uma taxa mais lenta. Uma explicação, fornecida por Wellens (2012), é que o MNG limita a capacidade de memória do agente, removendo os nomes do adversário em vez de realizar uma inibição gradual. A estratégia BLI, no entanto, tem os melhores resultados, uma vez que não remove todos os nomes adversários e promove reforços lineares nas pontuações, ao invés de possuir valores interpolados³.

Nossos resultados (Figura 29) são muito semelhantes aos resultados apresentados por Wellens (2012), indicando que o ESOM pôde atuar acoplado ao mecanismo de simulação mental, mesmo implementando um jogo de adivinhação (Wellens (2012), neste experimento, implementou um jogo de nomeação). É importante salientar essa diferença, pois nossa implementação possui uma estrutura bem mais robusta do que aquela de um jogo de nomeação, envolvendo três camadas cognitivas: percepção, categorização e linguagem. Ressaltamos que, como nossos agentes estavam em um ambiente dinâmico com propriedades desconhecidas *a priori*, as regiões semânticas dos domínios se formaram gradualmente enquanto participavam dos jogos. O ESOM, então, permitiu que as propriedades pudessem ser instanciadas e atuassem como um modelo interno do ambiente para cada agente, servindo para fundamentar os símbolos linguísticos utilizados durante as interações dos agentes. No final dos jogos, cada agente tinha 5 propriedades distintas, resultantes dos pontos de protótipo de seu ESOM. Portanto, podemos concluir que neste experimento o ESOM cumpriu o propósito desejado, segmentando o espaço conceitual em tempo de execução. As regiões semânticas descobertas serviram para instanciar propriedades que foram empregadas para fundamentar os nomes usados para descrevê-las. Embora cada agente tivesse suas representações categóricas, eles chegaram a um consenso linguístico nos três testes realizados, o que nos permite concluir também que houve a interoperabilidade semântica entre eles.

7.2 Experimento 2: Validação do Mecanismo de Simulação Mental Atuando nos Modos Percepção e Linguagem

Este experimento teve como objetivo principal verificar a capacidade do mecanismo de simulação mental proposto em prover a emergência de símbolos fundamentados em agentes artificiais. Na nossa concepção, essa capacidade decorre do fato do mecanismo empregar três de seus processos: a percepção de eventos, a nomeação de eventos e a interpretação de símbolos linguísticos, sejam eles composicionais ou não. Sendo assim, neste experimento validamos o desempenho do mecanismo de simulação atuando nos modos de percepção e linguagem.

Essa validação foi feita a partir do desenvolvimento de uma aplicação com-

³ Para uma explicação detalhada, consulte Wellens (2012), subseção 2.4.1.

putacional que simula um jogo de linguagem genérico e baseado em eventos. O jogo de linguagem torna-se um caso de uso interessante para os propósitos desse experimento, pois seu protocolo aplica os processos que desejamos avaliar.

Como visto no capítulo 3, em um jogo de linguagem há uma comunidade de agentes e um conjunto de objetos. Em diferentes instantes de tempo, os agentes seguem um protocolo de comunicação para estabelecer atos linguísticos a fim de nomear objetos e eventos ocorridos no ambiente. Embora os agentes não estejam realizando uma disputa entre si, no nosso trabalho nomeamos os atos linguísticos de duelos. Esse termo foi empregado para definir a execução de uma rodada do protocolo de comunicação que os agentes estabelecem entre si.

Assim, ao iniciar um duelo, um contexto é gerado no ambiente e dois agentes são escolhidos - um para fazer o papel de agente falante e outro para fazer o papel de agente ouvinte. O falante escolhe um objeto, imagina a execução de um evento sobre ele e aplica o processo de nomeação para gerar um símbolo de alta ordem para expressar esse evento. O ouvinte recebe o símbolo e aplica o processo de interpretação. O resultado de sua interpretação é a execução do evento no ambiente. Por sua vez, o falante aplica o processo de percepção a fim de gerar as categorias que formam o evento executado pelo ouvinte. Dessa forma ele consegue verificar se o evento percebido é semelhante àquele que ele imaginou. Em caso positivo, haverá reforço nas estruturas que mantêm as associações entre os símbolos e as categorias. Caso negativo, haverá a correção do comportamento do ouvinte.

A ideia que permeia um jogo é fazer com que uma comunidade de agentes, ao estabelecer esse protocolo e aplicar esses processos, consiga chegar a um consenso a respeito dos símbolos empregados para nomear os objetos e eventos do ambiente. Quando isso ocorre, dizemos que houve alinhamento linguístico e, conseqüentemente, podemos afirmar que a população está alinhada semanticamente. É por esse motivo que o jogo de linguagem foi escolhido como caso de uso de validação do mecanismo de simulação mental proposto.

Diversos cenários de jogos de linguagem foram configurados para este propósito. Eles intercalam diferentes tamanhos de populações de agente, quantidades e objetos distintos e o emprego de símbolos isolados e composicionais. O objetivo de se empregar uma diversidade de configurações foi o de exercitar o mecanismo de simulação em condições com diferentes graus de dificuldade, indo do mais simples para os mais complexos. Para avaliar o desempenho do *framework* (isto é, o desempenho dos agentes que o emprega), diversas métricas mencionadas no Capítulo 3 foram calculadas enquanto o jogo se desenrolava. Os resultados poderão ser conferidos ao final deste capítulo.

A aplicação computacional desenvolvida contou com algumas simplificações. Neste caso, o processo de descobrir categorias não foi exercitado neste experimento. Sendo

assim, algumas categorias que são empregadas nas categorizações realizadas durante o processo de percepção foram criadas a priori. Falaremos sobre elas mais adiante. Essa simplificação não comprometerá a análise de desempenho do mecanismo de simulação mental, pois o experimento 1 foi proposto para avaliar exclusivamente esse processo.

Para que fosse possível conferir a evolução do jogo que está sendo estabelecido, foi desenvolvida uma interface gráfica informativa. Ela dispõe de vários recursos que permitem visualizar o rastreamento realizado por cada agente durante o processo de percepção, o estado atual do duelo do jogo (seja a nomeação, interpretação, ou outros), a convergência dos símbolos isolados e das regras empregadas para combiná-los em sentenças e muitas outras informações.

A seguir apresentaremos mais detalhes sobre as categorias ontológicas modeladas a priori, o protocolo do jogo de linguagem desenvolvido, a interface gráfica de configuração, visualização e monitoramento do jogo, bem como também o resultado dos diversos cenários de jogos de linguagem que foram configurados para validação do mecanismo de simulação mental.

7.2.1 Categorias Ontológicas Modeladas para o Jogo

O ambiente simulado é composto por agentes (chamados de “pessoas” na interface do jogo), capazes de executar ações, figuras geométricas (que representam os objetos sobre os quais as ações são executadas) e um ringue (local onde os duelos do jogo ocorrem). Os agentes podem executar ações sobre as figuras geométricas de diferentes maneiras.

As categorias ontológicas são usadas para identificar os componentes da simulação a partir das informações brutas sensoreadas. Elas são, especialmente, necessárias para o processo de percepção executado pelos agentes. Apesar dessas categorias poderem ser geradas dinamicamente por métodos como o ESOM, descrito na seção 7.1, para os propósitos deste trabalho algumas delas foram criadas a priori (propriedades, modificadores, categorias de objetos e *affordances*). Já objetos conceituais, eventos e episódios são criados enquanto o jogo se desenvolve. Além de serem necessárias para percepção, as categorias ontológicas também são utilizadas nos processos de nomeação e interpretação, como descrito na seção 6.1.

As categorias criadas a priori são apresentadas na Figura 30. Cada uma é armazenada em um atributo no código Java, e o nome do atributo é utilizado no presente texto como identificação da categoria. Seus pontos e vetores prototípicos (*raw value groups*) podem ser conferidos no Anexo C deste trabalho.

Categorias de Objetos: ▪ <peessoa> ▪ <figura geométrica> • <triângulo> • <retângulo> • <quadrado> • <trapézio> • <círculo>	Propriedades: ▪ Domínio Posição • <norte> • <sul> • <leste> • <oeste> • <neutro> ▪ Domínio Direção • <para cima> • <para baixo> • <para esquerda> • <para direita> ▪ Domínio Cor • <vermelho> • <amarelo> • <verde> • <ciano> • <roxo> • <preto> • <escuro> • <claro> ▪ Domínio Forma • <trilateral> • <quadrilateral> • <peessoa> • <ringue> • <alto> • <sem borda> • <largo> • <enclinado> ▪ Domínio Tamanho • <pequeno> • <grande> • <neutro> •		Modificadores: ▪ Domínio Posição • <para cima> • <para baixo> • <para esquerda> • <para direita> ▪ Domínio Cor • <de vermelho> • <de amarelo> • <de verde> • <de ciano> • <de roxo> • <escurecer> • <clarear> ▪ Domínio Tempo • <lentamente> • <rapidamente>
Affordances: ▪ <mover> ▪ <pintar>			

Figura 30 – Lista de Categorias Ontológicas empregadas para a categorização dos componentes da simulação.

7.2.2 Protocolo do Jogo de Linguagem Implementado

Para validação do mecanismo de simulação mental proposto por este trabalho, foi implementado um jogo de linguagem genérico que emprega parte de seus processos e funcionalidades nas etapas que compõem seu protocolo. Assim, os agentes são capazes de sensorar o ambiente e extrair as informações brutas dos objetos que ele contém. Através do processo de percepção, eles conseguem categorizar esses objetos e identificar os eventos que lhes são aplicados. A comunicação entre eles pode transcorrer sobre eventos ou sobre os elementos categóricos que os compõem (agente, paciente, ação, qualidades de objetos e de ações). No momento da expressão, um agente pode escolher o que falar e o processo de nomeação se encarregará de gerar um símbolo de alta ordem (seja ele isolado ou composicional) para tal. O agente que ouve este símbolo deve tentar interpretá-lo. Nesse caso, o processo de interpretação também lhe provê essa capacidade.

Além das capacidade de percepção, nomeação e interpretação, o mecanismo de simulação mental também fornece ao jogo estratégias de aprendizado que permitem o reforço ou a penalização das associações entre os símbolos e categorias e regras sintáticas. Tais estratégias são fundamentais para que a língua possa evoluir.

O jogo implementado requer um conjunto de objetos $o_i \in O$ e uma população de agentes (onde, $a_i \dots a_n \in A$). Os agentes duelam linguisticamente sobre os objetos que compõem um contexto. Vale ressaltar que os agentes não estabelecem, de fato, uma “disputa” entre si. A palavra duelo foi aqui empregada somente para ressaltar uma interação linguística, que segue um protocolo, onde um agente expressa algo por meio de

um símbolo e o outro agente tenta interpretá-lo. O jogo é formado por diversos duelos desse tipo.

A cada novo duelo, um contexto no ambiente é gerado e dois agentes são escolhidos da população para desempenhar os papéis de falante e ouvinte. Ambos os agentes são confrontados com o mesmo contexto, que é composto de n objetos. Quando essas condições são atendidas, o duelo inicia-se, seguindo as etapas do protocolo apresentado na Figura 31. Para facilitar o entendimento de seu funcionamento, a execução do protocolo foi dividida em dois cenários distintos, como apresentado a seguir.

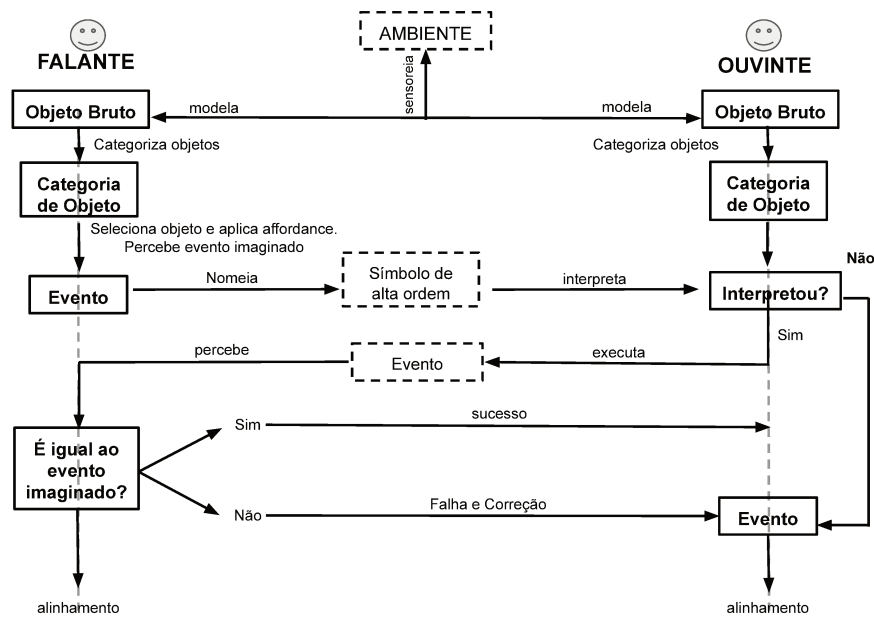


Figura 31 – O protocolo do jogo de linguagem implementado. Os quadrados representam os dados produzidos pelos agentes. Os textos referem-se aos processos inerentes a cada fase. Os processos podem falhar. Entretanto, existem estratégias de diagnóstico e reparo associadas a cada um deles. Um jogo de linguagem consiste em milhares dessas interações/duelos.

Cenário A

O cenário A descreve o protocolo executado quando o ouvinte não é capaz de interpretar o símbolo de alta ordem emitido pelo falante.

1. Falante e Ouvinte sensoreiam o contexto do ambiente;
2. Falante e Ouvinte categorizam os objetos;
3. Falante seleciona um objeto e imagina a execução de uma *affordance* sobre este objeto;
4. Falante categoriza um evento (esta etapa envolve o processo de percepção de eventos);

5. Falante gera um símbolo de alta ordem para expressar os elementos categóricos do evento que foram previamente configurados na interface do jogo (esta etapa envolve o processo de nomeação de eventos);
6. Ouvinte recebe o símbolo de alta ordem, o decompõe e tenta interpretá-lo (esta etapa envolve o mecanismo de interpretação de símbolos);
7. Ouvinte não conseguiu realizar a interpretação;
8. Falante penaliza a associação (símbolo, categorias) / (categoria, símbolos) empregada (esta etapa envolve a estratégia de penalização do aprendizado gramatical);
9. Duelo é finalizado e os dois agentes retornam às suas posições iniciais e um novo duelo pode ser iniciado.

Cenário B

O cenário B descreve o protocolo executado quando o ouvinte é capaz de interpretar o símbolo de alta ordem emitido pelo falante.

1. Falante e Ouvinte sensoreiam o contexto do ambiente;
2. Falante e Ouvinte categorizam os objetos;
3. Falante seleciona um objeto e imagina a execução de uma *affordance* sobre este objeto;
4. Falante categoriza um evento (esta etapa envolve o processo de percepção de eventos);
5. Falante gera um símbolo de alta ordem para expressar os elementos categóricos do evento que foram previamente configurados na interface do jogo (esta etapa envolve o processo de nomeação de eventos);
6. Ouvinte recebe o símbolo de alta ordem, o decompõe e tenta interpretá-lo (esta etapa envolve o mecanismo de interpretação de símbolos);
 - a) Ouvinte interpretou todos os símbolos;
 - b) Ouvinte conseguiu interpretar apenas alguns dos símbolos:
 - i. Ouvinte completa o evento com elementos categóricos aleatórios para conseguir interpretá-lo na totalidade;
7. Ouvinte executa o evento gerado a partir da interpretação feita;
8. Falante analisa o evento executado pelo ouvinte;

- a) O evento executado pelo ouvinte é igual ao evento imaginado pelo falante:
 - i. Falante sinaliza sucesso;
 - ii. Ouvinte e Falante reforçam as associações (símbolo, categorias) / (categoria, símbolos) empregadas (Esta etapa envolve a estratégia de reforço do aprendizado gramatical);
 - b) O evento executado pelo ouvinte é diferente do evento imaginado pelo falante:
 - i. Falante executa evento de correção;
 - ii. Ouvinte percebe evento de correção e realiza a categorização de seus elementos;
 - iii. Ouvinte associa as categorias aos símbolos que foram expressos e insere as novas associações nos dicionários lexicais (árvore e tabela);
 - iv. Ouvinte penaliza as associações (símbolo, categorias) / (categoria, símbolos) que ocasionaram o erro de interpretação (esta etapa envolve a estratégia de penalização do aprendizado gramatical);
 - v. Falante e ouvinte reforçam as associações (símbolo, categorias) / (categoria, símbolos) que geraram a interpretação correta (esta etapa envolve a estratégia de reforço do aprendizado gramatical);
9. Se o ouvinte foi capaz de identificar a ordem dos elementos do símbolo de alta ordem criado pelo falante, o mesmo registra essa ordenação (esta etapa envolve a estratégia de reforço do aprendizado sintático);
10. O duelo é finalizado e os dois agentes retornam às suas posições iniciais e um novo duelo pode ser iniciado.

Sobre o protocolo descrito, vale ressaltar que o reforço e a penalização das associações entre os símbolos e as categorias são executados conforme a estratégia de aprendizado descrita na subseção 6.1.2. Já o registro da ordem dos elementos da sentença segue a estratégia descrita na seção 6.2. Durante a etapa de correção, o ouvinte deve perceber e categorizar o evento executado pelo falante. Quando um agente não consegue detectar um evento, o duelo é cancelado, os agentes voltam às suas posições iniciais e um novo duelo pode ser inicializado.

Repare que afirmamos aqui que o protocolo implementado é genérico. Isso significa que ele permite com que diferentes tipos de jogos de linguagem possam ser estabelecidos. Como será apresentado na próxima seção, a interface gráfica que foi desenvolvida para controle e depuração do jogo de linguagem contém uma área de configuração, onde é possível informar quais elementos categóricos que compõem o evento devem ser nomeados e expressos. Dependendo da configuração informada, o protocolo admitirá jogo de nomeação, de ação ou composicionais. Então, por exemplo, se na interface for informado

que o falante deve nomear somente a ação que compõe o evento, um jogo de ação será estabelecido. Se for informado que o falante deve nomear somente o paciente do evento, um jogo de nomeação será estabelecido. Por outro lado, se for informado que somente o paciente do evento deve ser nomeado e, será discriminado por meio de suas propriedades, um jogo de linguagem composicional, que emprega sentenças nominais, se estabelecerá. Se, entretanto, forem configurados todos os elementos possíveis de serem nomeados, um jogo de linguagem composicional também será estabelecido, porém, sentenças verbais serão empregadas.

Assim, caso a configuração do jogo especifique que somente o nome dos agentes deve ser comunicado, o protocolo difere um pouco daquele que foi apresentado na figura 31. Nesse caso, o protocolo seguido é:

1. O falante verbaliza o próprio nome, escolhendo um novo nome caso ainda não possua;
2. O ouvinte reforça a associação do nome recebido com o objeto conceitual do falante;
3. Com o duelo finalizado, os dois agentes retornam às suas posições iniciais e um novo duelo pode ser iniciado.

7.2.3 Interface Gráfica

Para que fosse possível configurar, controlar e depurar a execução do jogo de linguagem implementado, foi desenvolvida uma interface gráfica informativa. Como mostra a Figura 32, ela é composta de quatro áreas distintas.

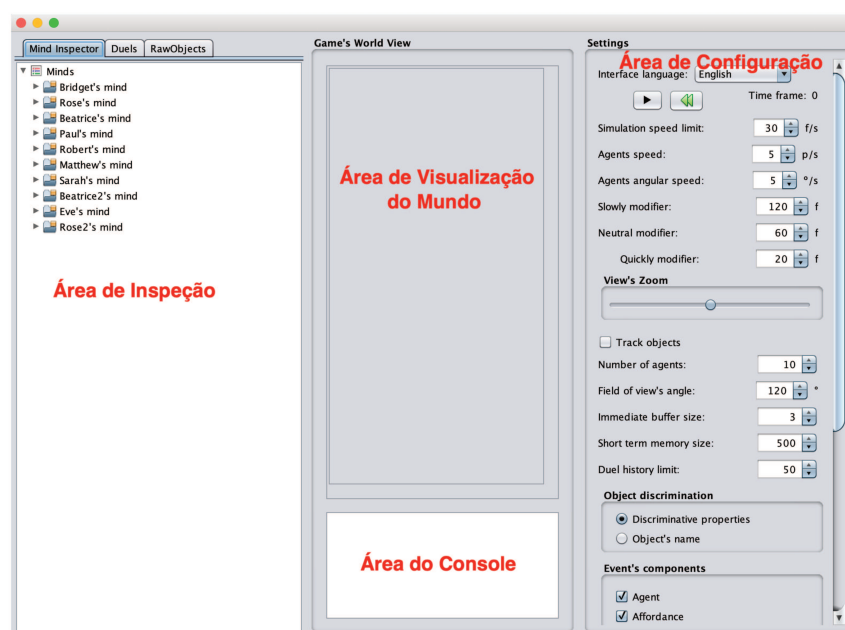


Figura 32 – Interface da Aplicação do Jogo de Linguagem.

A área de visualização do mundo é o local onde a execução do jogo pode ser visualizada. A área de configuração é onde devem ser informadas as configurações desejadas para a execução do jogo. A área de inspeção foi criada para consultar a mente dos agentes, os duelos estabelecidos e os objetos no mundo. Por fim, a área de console foi criada com o objetivo de visualizar as sentenças emitidas pelo agente falante durante o duelo. Para detalhes adicionais a respeito de cada área mencionada, esses podem ser consultados no Anexo C, na seção C.2.

A fim de ilustrar a maneira como os recursos implementados na interface podem nos servir durante a depuração, foi executado um jogo de linguagem e extraídos diversos *prints* com informações a respeito de cada etapa do protocolo.

7.2.3.1 Exemplo de Jogo de Linguagem para Ilustrar Recursos da Interface Gráfica

O jogo de linguagem configurado neste exemplo é um jogo composicional, consistindo de um ambiente composto por 2 objetos da categoria círculo e uma população de 2 agentes. Sentenças verbais, compostas por: agente, paciente, propriedades, ação e modificadores, foram empregadas para descrever eventos aplicados sobre os objetos. Como o ambiente contém objetos pertencentes a uma mesma categoria, o paciente (objeto que sofrerá a ação) será identificado por meio de suas propriedades discriminativas. Vejamos mais detalhes a seguir.

Para facilitar a depuração, nomes foram atribuídos aos agentes. Assim sendo, um deles foi chamado de Adriano e o outro de Tiago. A Figura 33 mostra que eles iniciaram um duelo (o primeiro do jogo), sendo Adriano o agente falante e Tiago o ouvinte.

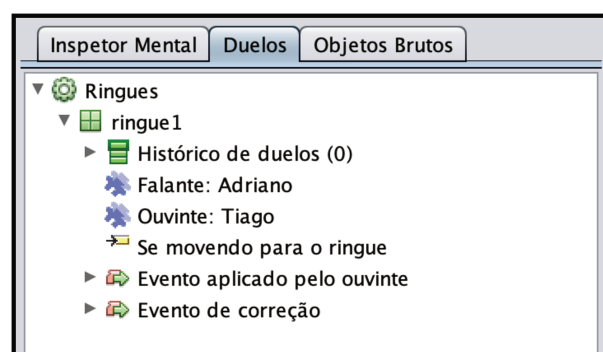


Figura 33 – Aba de inspeção de duelos - Início de um novo duelo linguístico entre Adriano (falante) e Tiago (ouvinte)

Os agentes sensoriam o ambiente e criam em suas mentes a representação bruta dos objetos. Essa representação é usada para categorizar os objetos. O resultado dessa ação é a criação de uma categoria de objeto. Este é o primeiro passo do processo de percepção, que na sequência identifica cada categoria, criando seus respectivos objetos conceituais. O algoritmo de rastreamento é empregado para montar na memória de curto

prazo a trajetória temporal de cada objeto conceitual identificado. Para isso, é associada a cada objeto conceitual sua representação bruta gerada a partir dos diferentes *frames* em que o objeto foi sensoreado. A Figura 34 apresenta: (a) o objeto “círculo 1” presente no ambiente, (b) sua representação bruta gerada pelo processo de modelagem de informações e, (c) por fim, seu objeto conceitual, criado pelo processo de percepção e o início da construção de sua trajetória temporal. Repare que no ambiente o objeto é identificado como “círculo 1”, mas na mente ele é identificado como um objeto conceitual de nome “círculo 2” (esse nome é apenas uma designação interna, não tendo uma correlação direta com o nome utilizado pelo jogo no ambiente), assumindo as propriedades: tamanho “pequeno”, posição “leste e norte”, formato “sem arestas/bordas” e cor “azul, saturado e brilho neutro”.

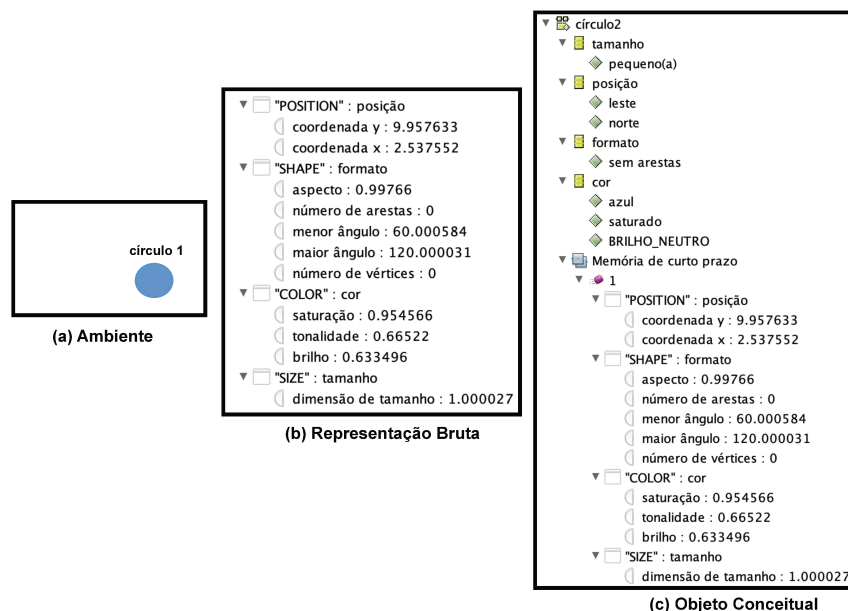


Figura 34 – Os diferentes níveis de representação mental gerado pelo mecanismo de simulação mental no jogo de linguagem.

Na sequência do protocolo, o falante, escolhe e aplica mentalmente (ou seja, imagina-se aplicando) uma *affordance* sobre um objeto conceitual escolhido. O resultado dessa ação corresponde a um evento (nesse momento, só imaginado). Ele emprega o processo de nomeação, que cria uma sentença verbal, para expressar esse evento imaginado. A Figura 35 apresenta a sentença verbal gerada, bem como também as categorias que formam o evento imaginado.

Como pode ser observado, cada símbolo que compõe a sentença expressa as seguintes categorias ontológicas:

- quenoqu - o modificador (do domínio do tempo) lentamente.
- welzolybilmi - a categoria de objeto círculo (neste caso, o círculo 1).

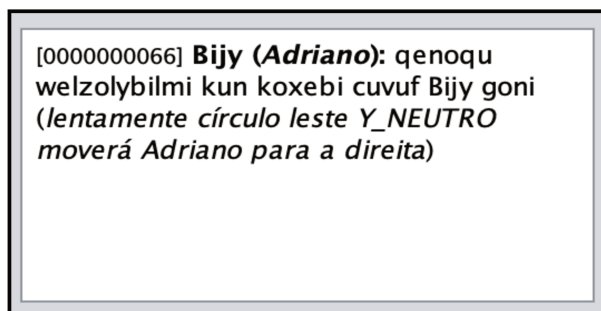


Figura 35 – Sentença Expressada pelo Falante.

- kun - a propriedade (do domínio da posição) leste;
- koxebi - o modificador (do domínio da posição) Y NEUTRO, indicando que a *affordance* vai ser aplicada mantendo a coordenada Y intacta.
- cuvuf - a *affordance* mover.
- Bijy - o objeto conceitual Adriano.
- goni - o modificador (do domínio da posição) para direita.

Vale ressaltar, que a categoria associada a um símbolo é que define seu papel gramatical.

Por outro lado, pode-se entender também que foram aplicadas as seguintes regras sintáticas para estruturar a sentença:

- Regra Sintática 1 (a que define a posição do agente em relação ao verbo): Agente após o verbo”.
- Regra Sintática 2 (a que define a posição de uma propriedade em relação ao agente/paciente): Propriedade (do domínio da posição) após agente/paciente;
- Regra Sintática 3 (define a posição de modificador em relação as 4 posições possíveis): Modificador (do domínio da posição) na 2a. ou 4a. posição (ou seja, antes do verbo ou no final da sentença); modificador (do domínio do tempo) na 1a. posição.
- Foi aplicado a transformação de tempo futuro no símbolo que designa o verbo.

Ao receber essa sentença, o ouvinte tenta interpretá-la. Porém, como seus dicionários lexicais estão vazios, ele não consegue gerar a interpretação. Isso ocorreu porque se trata do primeiro duelo do jogo e os agentes ainda estão aprendendo. Sendo assim, ele sinaliza para o falante que não foi capaz de interpretar a sentença informada, pois não conhece nenhum símbolo que a formou. O falante, após receber essa informação, inicia o processo de correção.

Esse processo consiste no falante executando o evento no ambiente, para que o ouvinte possa percebê-lo e fazer as devidas associações, entre categoria e símbolos, em seus dicionários.

Enquanto o falante executa sua performance, o ouvinte está sensoreando o ambiente. Ele, por meio do processo de percepção, descobre que o objeto conceitual alvo é o círculo 2. Então, o ouvinte registra suas informações brutas na memória de curto prazo, continuando, assim, o rastreamento desse objeto. A Figura 36, apresenta a memória de curto prazo do ouvinte, após o registro das informações brutas provenientes dessa etapa do protocolo.

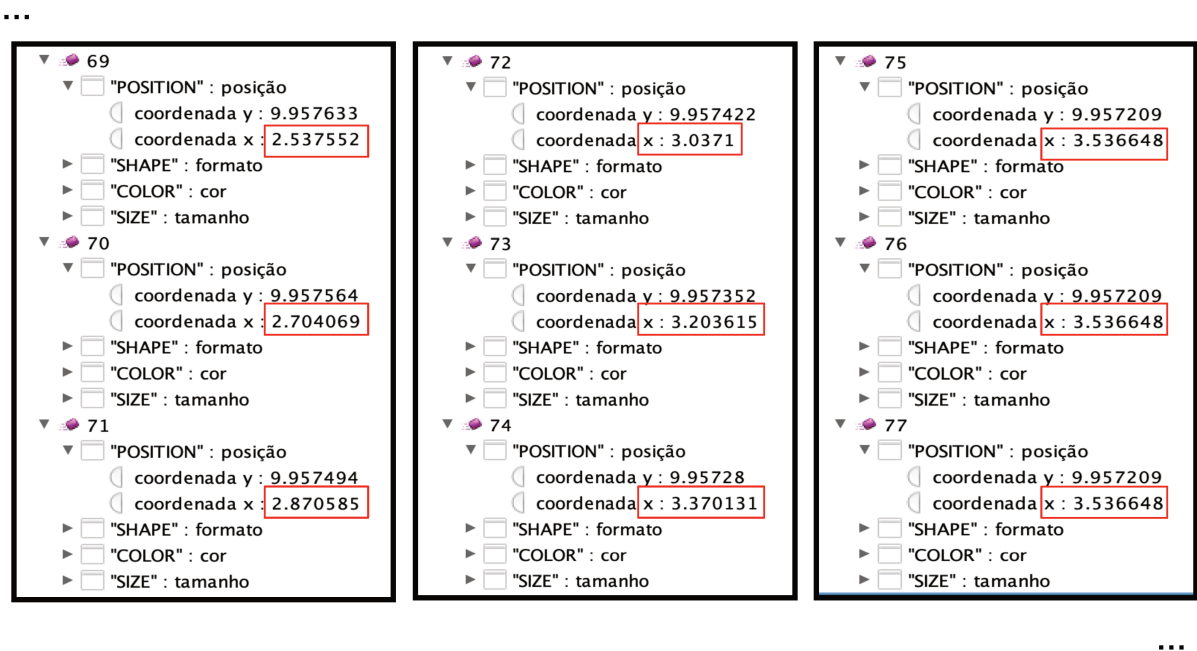


Figura 36 – O rastreamento do objeto conceitual círculo 2 na mente do ouvinte.

A todo momento, as *affordances* modeladas no mecanismo de simulação mental analisam os objetos brutos mantidos na memória de curto prazo, em busca de algum padrão de alteração que possa sinalizar que alguma delas foi aplicada. No caso apresentado na Figura 36 verificou-se que a coordenada X está sendo alterada. Esse fato, portanto, representa que *affordance* mover foi aplicada, o que dá início à construção do evento, como mostra a Figura 37.

Repare que no início da construção do evento, o ouvinte conseguiu categorizar o agente, o paciente e a *affordance* do evento. Entretanto, o mesmo não aconteceu para os modificadores. Isso acontece porque o evento ainda não foi finalizado. Somente próximo ao final, que o ouvinte tem condições de categorizar corretamente a maneira como a *affordance* se desenrolou. Repare também que a construção do evento ocorreu a partir do *frame* 69 e finalizou no *frame* 75, quando não houve mais alteração da coordenada X.

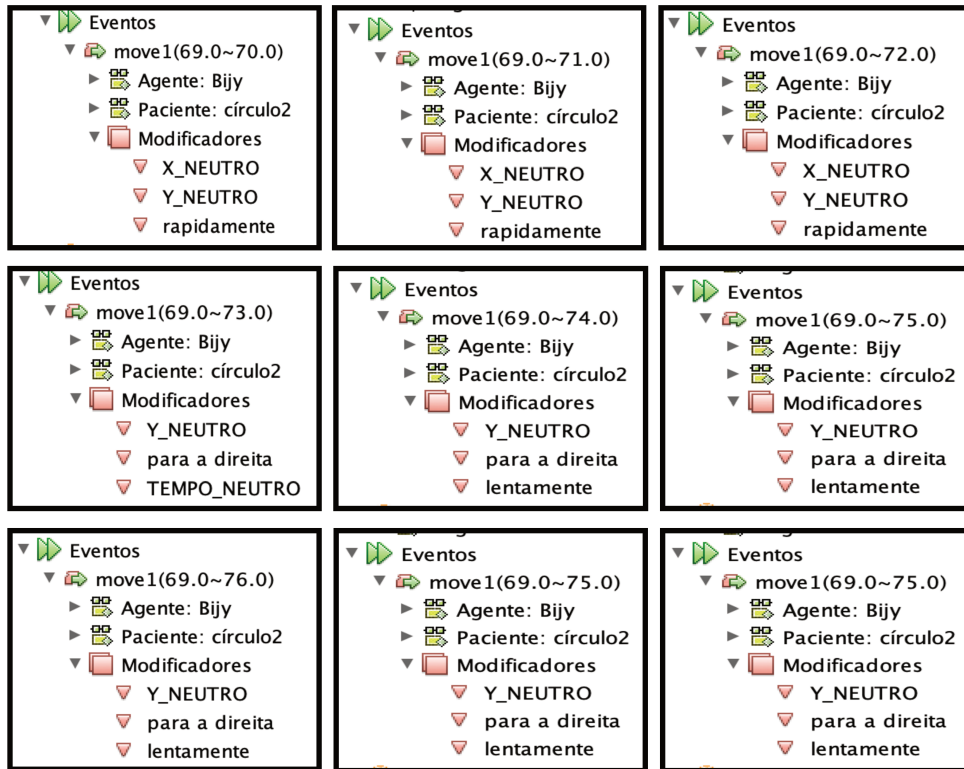


Figura 37 – A construção do evento na mente do ouvinte.

A próxima etapa do protocolo do jogo consiste no ajuste dos dicionários lexicais por parte do ouvinte e falante, o que ocorre através das estratégias de aprendizado gramatical e sintático. No caso do falante, ele vai penalizar os símbolos utilizados, pois não houve sucesso comunicativo. No caso do ouvinte, ele vai adotar os símbolos recebidos. Porém, existe um problema nessa adoção.

Como visto, o ouvinte extraiu, do processo de correção, os objetos brutos sensorados e as categorias ontológicas derivadas do processo de percepção. Mas, como ele não sabe qual categoria cada símbolo designa (ou qual categoria fundamenta o significado de cada símbolo), ele associa todos os símbolos recebidos a cada uma das categorias encontradas (isso na tabela de categorias). Da mesma forma, na árvore de símbolos, ele associa todas as categorias percebidas a cada um dos símbolos recebido. Em razão da indecisão a respeito dos papéis gramaticais dos símbolos, o ouvinte não consegue abstrair as regras empregadas na construção da sentença recebida, não havendo neste caso aprendizado sintático. O duelo é finalizado, os agentes retornam para suas posições iniciais e um novo duelo pode se iniciar.

7.2.3.2 A Evolução da Linguagem durante o Jogo de Linguagem

Leva-se um tempo para que os agentes cheguem a um consenso a respeito dos símbolos empregados para designar as categorias (e qual categoria vai fundamentar um

símbolo) e sobre as regras utilizadas para estruturar as sentenças. Quando isso acontece dizemos que a população alcançou interoperabilidade semântica. Entretanto, até o alinhamento ocorrer, sucessivos duelos devem ser estabelecidos para que os agentes possam reforçar, punir e adotar associações entre categorias e símbolos e aprender regras sintáticas.

Para ilustrar a evolução do aprendizado gramatical e sintático, continuamos a execução do jogo apresentado nos parágrafos anteriores. Deixamos que vários duelos fossem estabelecidos ao longo do processamento de 2000 *frames*. A Figura 38 apresenta como ficou a linguagem de um dos agentes após esse processamento.

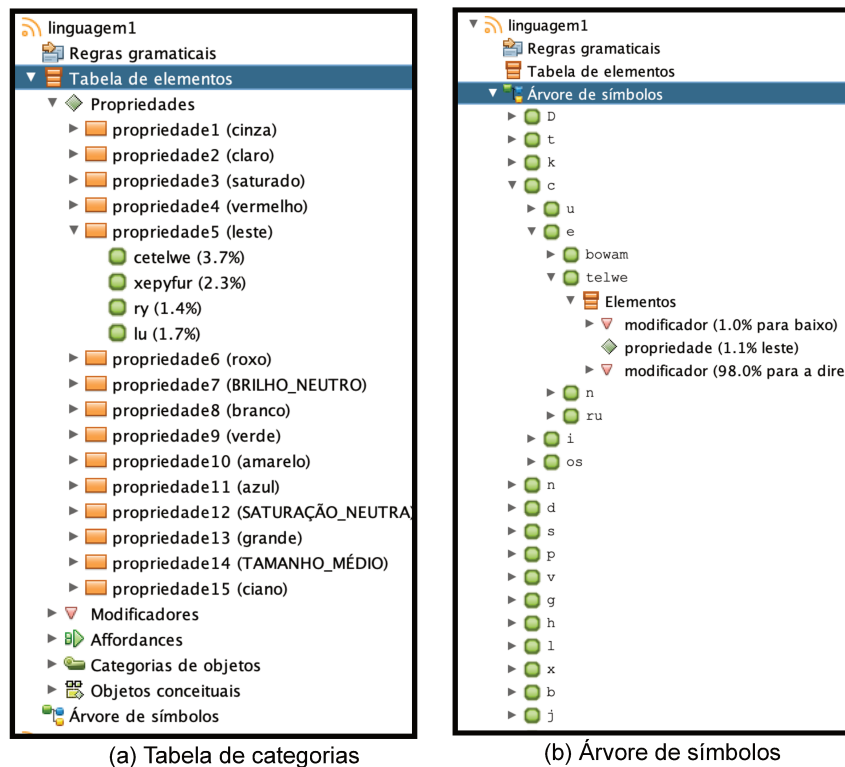


Figura 38 – A linguagem de um dos agentes após o processamento de 2000 *frames*.

Na tabela (a) é possível verificar que os agentes estão aprendendo a nomear vários tipos de categorias (propriedades, modificadores, *affordances*, categorias de objetos e objetos conceituais). Pegamos como exemplo a parte que exibe as propriedades, onde há uma variedade delas. Isso ocorre porque as propriedades são empregadas pelos agentes para discriminar os dois tipos de círculos que há no ambiente. Para o caso da propriedade *leste* (do domínio da posição) há 4 símbolos competindo para nomeá-la. Neste caso, o nome **cetelwe** é o que tem maior probabilidade, sendo, naquele instante, o mais provável de ser empregado pelo processo de nomeação. Porém, ao analisar a árvore de símbolos (b), percebemos que existem três categorias disputando para fundamentar o significado deste símbolo. Neste caso, a propriedade *leste* está com o menor percentual e quem está com o maior é o modificador *para a direita*. Muito provavelmente, o símbolo **cetelwe**

não será o símbolo convencionalizado para designar a propriedade leste. Como ainda não houve o aprendizado gramatical, ainda há muita disputa entre símbolos e categorias.

Deixamos o jogo transcorrer até o processamento de 10000 *frames*, e o estado da linguagem neste ponto do jogo pode ser conferido na Figura 39.

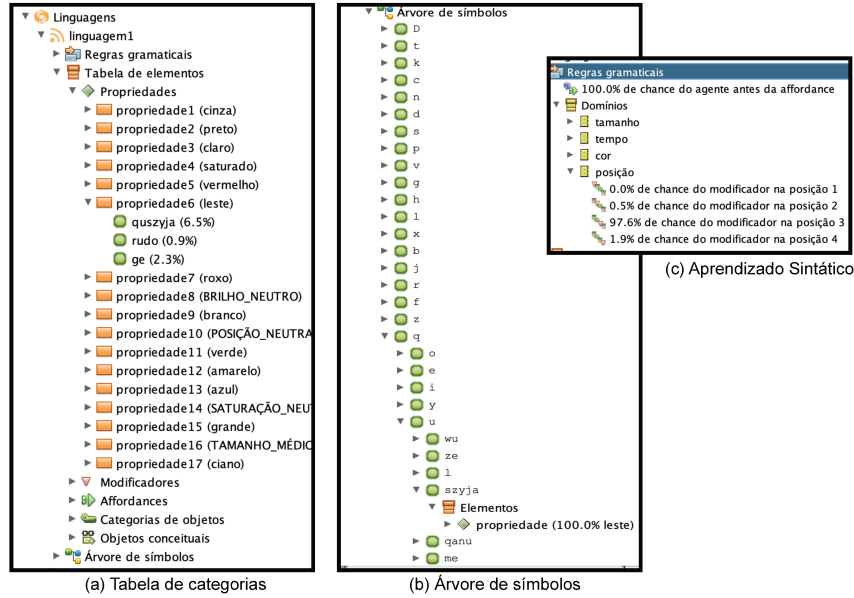


Figura 39 – A linguagem de um dos agentes após o processamento de 10000 *frames*.

Na tabela de categorias (a), consultamos a propriedade leste novamente e, como era esperado, o símbolo **cetelwe** foi esquecido. Agora, o símbolo que tem maior probabilidade de nomear esta categoria é o símbolo **quszyja**, com 6,5% de probabilidade. Também consultamos esse nome na tabela de símbolos (b) e verificamos que já há uma convergência. A propriedade leste é a empregada para fazer a fundamentação deste símbolo. A pergunta que surge é: será que esse símbolo será adotado para nomear a propriedade leste? Analisando também o aprendizado sintático (c), percebemos que ele começou a se estabelecer. Com relação ao posicionamento do agente e do paciente em relação ao verbo, já é consenso de que o agente deve vir antes da *affordance* e, assim sendo, o paciente virá depois. Sobre o posicionamento das propriedades e modificadores do domínio da posição, percebemos que para os modificadores já é quase consenso de que eles ficarão na 3a. posição possível (após o verbo da sentença). Já no caso das propriedades, ainda não há uma definição.

Continuamos a execução do jogo, mas dessa vez deixamos que se processassem 50000 *frames*. Novos duelos aconteceram e, então, verificamos novamente o desenvolvimento da linguagem dos agentes. A Figura 40 apresenta o estado da linguagem nesse ponto da execução do jogo.

Na tabela de categorias (a), consultamos novamente a propriedade leste e percebemos que o símbolo **quszyja** também foi esquecido. O mesmo aconteceu na árvore

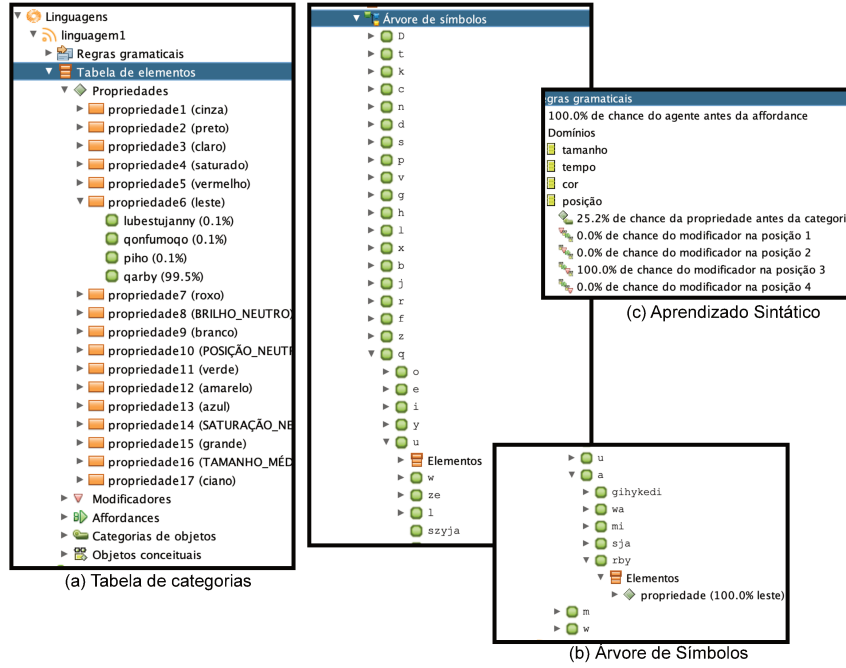


Figura 40 – A linguagem de um dos agentes após o processamento de 50000 *frames*.

de símbolos (b), reafirmando que ele nunca mais será empregado nos jogos.

Porém, desta vez, parece haver um consenso sobre qual símbolo e categoria empregar. Como uma probabilidade bem maior (99.5%), o símbolo **qarby** é agora o utilizado para designar a propriedade leste. Essa correlação é confirmada pela árvore de símbolos (b), onde, com 100% de probabilidade, a propriedade leste é quem faz a fundamentação deste símbolo. Portanto, pode-se concluir que os agentes chegaram a um consenso sobre qual nome utilizar para designar a propriedade leste.

Definições como essa afetam o aprendizado sintático (c), onde é possível verificar que os agentes já sabem empregar as propriedades do domínio da posição. Neste caso, há 25% de probabilidade desta ficar antes da categoria de objeto que designa o paciente. Ainda serão necessários mais alguns duelos para que haja completo alinhamento gramatical e, especialmente o sintático. Mesmo não exibindo os resultados finais aqui, afirmamos que o completo alinhamento (gramatical e sintático) ocorre perto do processamento de 250000 *frames*. Portanto, podemos concluir que para haver interoperabilidade semântica entre os agente que trocam símbolos de alta ordem em um jogo de linguagem, sucessivos duelos devem ocorrer.

7.2.4 Experimento 2: Resultados e Discussões

Para validar a aplicação do jogo de linguagem desenvolvida, três tipos de experimentos foram executados com diferentes tamanho de população. No primeiro deles (referenciado como T1) é executado um jogo de adivinhação, onde os agentes devem chegar

ao consenso sobre o nome das figuras geométricas dispostas no ringue. Já no segundo (referenciado como T2) os agentes devem executar um jogo de linguagem composicional e empregar o nome de uma figura geométrica juntamente com uma propriedade para discriminá-la das demais figuras pertencentes à mesma categoria. No terceiro e último experimento (referenciado como T3), os agentes também empregam jogos de linguagem composicionais, porém para descrever eventos ocorridos no ambiente.

Como forma de mensurar o desempenho da população foram extraídas como métricas o sucesso comunicativo e sucesso de alinhamento. Para os casos que se aplicaram, o sucesso lexical também foi mensurado. O leitor pode nos questionar se essas métricas por si só comprovariam a efetividade do mecanismo de simulação proposto por esse trabalho. E a resposta para esta pergunta é sim.

Para que uma população de agentes alcance resultados relevantes e significativos para essas métricas, ela depende de uma boa execução dos duelos estabelecidos ao longo do jogo. Assim sendo, o jogo depende do mecanismo de simulação mental, que provê os processos necessários para sua execução. De modo especial, podemos destacar o processo de percepção de eventos que realiza a representação icônica do ambiente, baseada em categorias ontológicas. Tal fato, confere aos agentes a capacidade de manipular tanto símbolos isolados quanto símbolos composicionais (sentenças de símbolos), todos eles fundamentados a partir das experiências que o agente estabelece com o ambiente em que está inserido. Portanto, se concluímos que houve interoperabilidade semântica entre os agente que trocam símbolos de alta ordem neste jogo de linguagem, estamos também validando o desempenho do *framework* de simulação que foi proposto. Esse é portanto, o objetivo do experimento apresentado nessa seção.

7.2.4.1 A Configuração Adotada

Antes de entrarmos em detalhe sobre cada experimento, vale ressaltar as configurações que são comuns entre eles. Os seguintes parâmetros foram aplicados na área de configuração da interface gráfica:

- **Para velocidade:**

- *Simulation speed limit*: 300 f/s;
- *Agents speed*: 500 p/s;
- *Agents angular speed*: 500°/s;
- *Modifiers*:
 - * *Slowly*: 6 f;
 - * *Neutral*: 4 f;

* *Quickly*: 2 f;

- **Para o tamanho das estruturas de memória:**

- *Immediate Buffer*: 3;
- *Short Term Memory*: 500;
- *Duel history limit*: 50;

- **Para a visão do agente:**

- *Field of view's angle*: 360°;

- **Para os símbolos isolados:**

- *Maximum number of syllables in a word*: 6;

Os seguintes parâmetros foram aplicados para a estratégia de aprendizado gramatical:

- Taxa de incremento (p_a): 1.0;
- Percentual mínimo de reforço (p_{mr}): 0.20;
- Percentual mínimo de penalização (p_{mp}): 0.03;
- Percentual de remoção (p_{rem}): 0.0005;
- Percentual para reduzir a penalização (p_{ne}): 0.8;

7.2.4.2 Cenário T1

Assim como em um jogo de adivinhação, o cenário T1 tem como objetivo principal categorizar símbolos isolados para nomear categorias de objetos e verificar se o mecanismo de aprendizado é capaz de garantir a interoperabilidade semântica nesse contexto. Trata-se do experimento mais simples que foi estabelecido .

O ambiente possui 1 ringue apenas, composto por 5 figuras geométricas distintas: 1 círculo, 1 triângulo, 1 retângulo, 1 quadrado e 1 trapézio. O único componente do evento que será expresso é o paciente. E como os objetos são de figuras distintas entre si, não será necessário o emprego de suas propriedades para discriminá-los.

Para validação deste experimento, em um primeiro momento foram executados jogos em uma população de 2 agentes e foram extraídas as métricas descritas na seção [3.3.2](#). No contexto do cenário T1, estas métricas são: Sucesso Comunicativo; Sucesso de

Alinhamento; Alinhamento de Nome (neste caso, da categoria de objeto). Foram executadas 10 repetições com uma média de 1100 duelos, a partir das quais para cada métrica calculou-se a média móvel simples de 100 duelos e seu desvio padrão. Os resultados apurados podem ser observados na Figura 41.

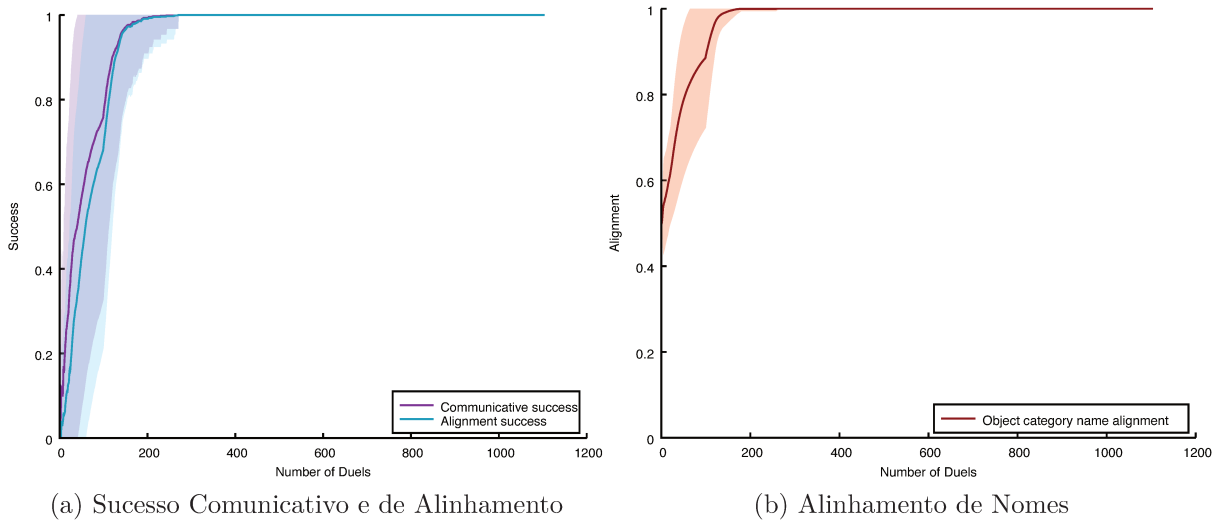


Figura 41 – Cenário T1 com 2 Agentes: Resultado experimental para estratégia de aprendizado gramatical descrita na seção 6.1.2. Parâmetros utilizados na estratégia: p_a : 1.0, p_{mr} : 0.20, p_{mp} : 0.03, p_{rem} : 0.0005, p_{ne} : 0.8. População de 2 agentes e ringue com 5 figuras geométricas distintas. Foram executados aproximadamente 1100 duelos. Em (a) os agentes alcançaram completo sucesso comunicativo e de alinhamento. Em (b) o alinhamento dos nomes empregados para as categorias de objetos também foi completo.

Como pode ser observado nos gráficos da Figura 41, houve completo sucesso comunicativo e de alinhamento. Isso indica que a estratégia implementada permitiu com que os agentes conseguissem interpretar os símbolos empregados para nomear as figuras geométricas e também chegassem a um consenso a respeito dos nomes empregados para tal. Como trata-se de uma população pequena (2 agentes), eles gastam em torno de 300 duelos para alcançar o pleno alinhamento e sucesso comunicativo.

O alinhamento de nomes iniciou o jogo com o percentual de 50%, pois no início do experimento cada agente tem preferência por um nome, mas ambos rapidamente alinham os nomes empregados para as categorias de objetos. Dada a simplicidade do cenário, após a convergência não ocorreu nenhuma oscilação, o que indica que os agentes estão alinhados semanticamente.

Como o desempenho de experimentos que envolvem jogos de nomeação (ou de adivinhação) é fortemente influenciado pelo tamanho da população, o mesmo foi repetido com a mesma configuração de ambiente, porém com uma população de 6 agentes, sendo extraídas as mesmas métricas (sucesso comunicativo, sucesso de alinhamento e alinhamento de nome).

Sendo assim, para uma população de 6 agentes foram executadas 10 repetições com uma média de 8500 duelos, onde também calculou-se a média móvel simples de 100 duelos e seu desvio padrão. Os resultados apurados podem ser observados na Figura 42.

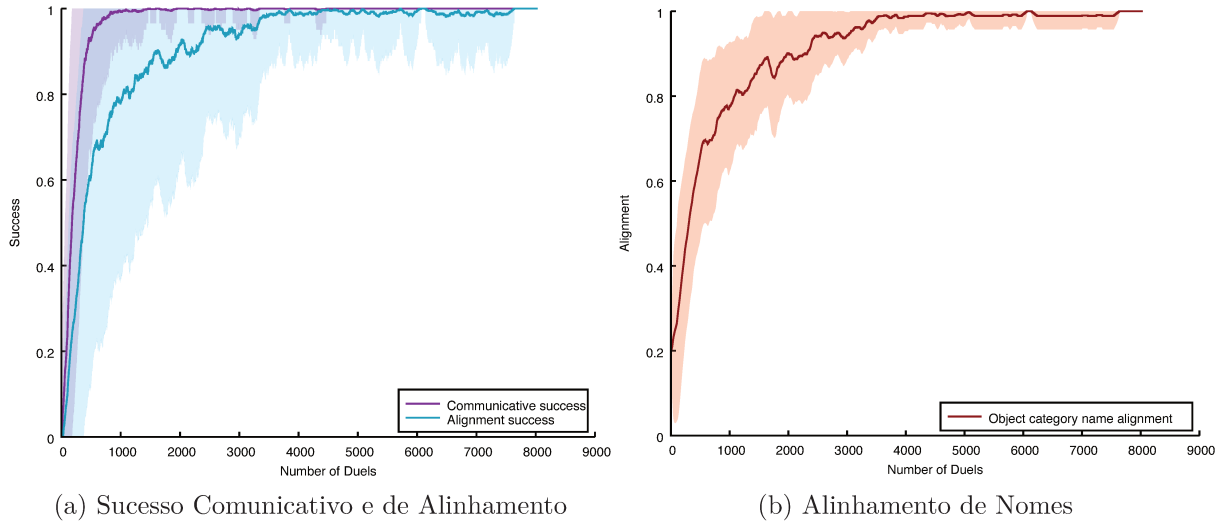


Figura 42 – Cenário T1 com 6 Agentes: Resultado experimental para estratégia de aprendizado gramatical descrita na seção 6.1.2. Parâmetros utilizados na estratégia: p_a : 1.0, p_{mr} : 0.20, p_{mp} : 0.03, p_{rem} : 0.0005, p_{ne} : 0.8. População de 6 agentes e ringue com 5 figuras geométricas distintas. Foram executados aproximadamente 8500 duelos. Em (a) os agentes alcançaram completo sucesso comunicativo e de alinhamento. Em (b) o alinhamento dos nomes empregados para as categorias de objetos também foi completo.

Os gráficos apresentados na Figura 42 comprovam que o desempenho do experimento é influenciado pelo tamanho da população. Em uma população de 2 agentes foram gastos cerca de 300 duelos para o completo sucesso comunicativo e de alinhamento. Com 6 agentes esse número aumentou consideravelmente, sendo necessários cerca de 8500 duelos.

Reparando no gráfico (a), antes do alinhamento, houve bastante oscilação em relação ao comportamento médio. Isso indica que quanto maior o tamanho da população, maior será a quantidade de símbolos disputando para nomear as figuras geométricas. Assim sendo, mais duelos precisam ser estabelecidos para que toda a população tenha preferência pelos mesmos nomes. Repare também que perto dos 4000 duelos a população de agentes está quase totalmente alinhada.

Como a população é composta por 6 agentes, o gráfico (b) do alinhamento de nome iniciou o experimento com 16.7% de alinhamento, e após aproximadamente 8000 duelos ele já estava em 100%. Como esta métrica também influencia o sucesso de alinhamento, após essa marca o mesmo também estava alinhado.

Executar o mesmo experimento com populações de tamanhos diferentes e al-

cançar completo sucesso comunicativo e de alinhamento mostra a efetividade do modelo de representação conceitual e do mecanismo de linguagem para o contexto proposto. Vejamos a seguir mais alguns experimentos que foram propostos para validar outras características do *framework*.

7.2.4.3 Cenário T2

O cenário T2 foi proposto para validar o desempenho do mecanismo de simulação mental no contexto da composicionalidade. Seu objetivo principal é fundamentar sentenças nominais criadas para distinguir objetos pertencentes a uma mesma categoria. Assim, são executados jogos de linguagem composicionais e espera-se que o *framework* computacional implementado garanta a interoperabilidade semântica também nesse contexto.

O ambiente deste experimento é composto por apenas um ringue que possui 8 figuras geométricas: 4 círculos e 4 triângulos. Como existem objetos de mesmas figuras, será necessário o emprego de suas propriedades para discriminá-los. Assim, embora seja expresso apenas o paciente do evento, a categoria do mesmo deverá vir acompanhada de propriedades, configurando a possibilidade da formação de sentenças composicionais nominais.

Nos primeiros testes executados foram estabelecidos jogos em uma população de 2 agentes e foram extraídas como métricas o sucesso comunicativo, o sucesso de alinhamento, o alinhamento de nome (neste caso, das categorias de objeto e das propriedades) e o alinhamento sintático. Semelhante ao cenário T1, foram executadas 10 repetições do cenário T2 com 2 agentes, a partir das quais se calculou a média móvel de 100 duelos e seu desvio padrão para cada métrica mencionada. Os resultados apurados podem ser observados na Figura 43.

Analisando o gráfico (a) presente na Figura 43, percebemos que houve completo sucesso comunicativo e de alinhamento. Esse resultado indica que os dois agentes da população conseguiram interpretar as sentenças nominais empregadas para descrever os objetos do ambiente e que chegaram a um consenso sobre os símbolos utilizados nessa tarefa. Embora o experimento tenha sido executado por aproximadamente 4000 duelos, após cerca de 2000 duelos a convergência já era aparente.

Repare que em alguns pontos, tanto para o sucesso comunicativo quanto para o de alinhamento, há oscilações no desvio padrão. Isso ocorre porque os agentes, mesmo preferindo os mesmos nomes para formar as sentenças (veja no gráfico (b)), devem possuir em seus dicionários lexicais (na árvore de símbolos e na tabela de categorias) nomes preteridos. Mesmo tendo uma probabilidade pequena, os nomes preteridos podem ser escolhidos para formar as sentenças. Quando isso ocorre, há erro comunicativo e, conse-

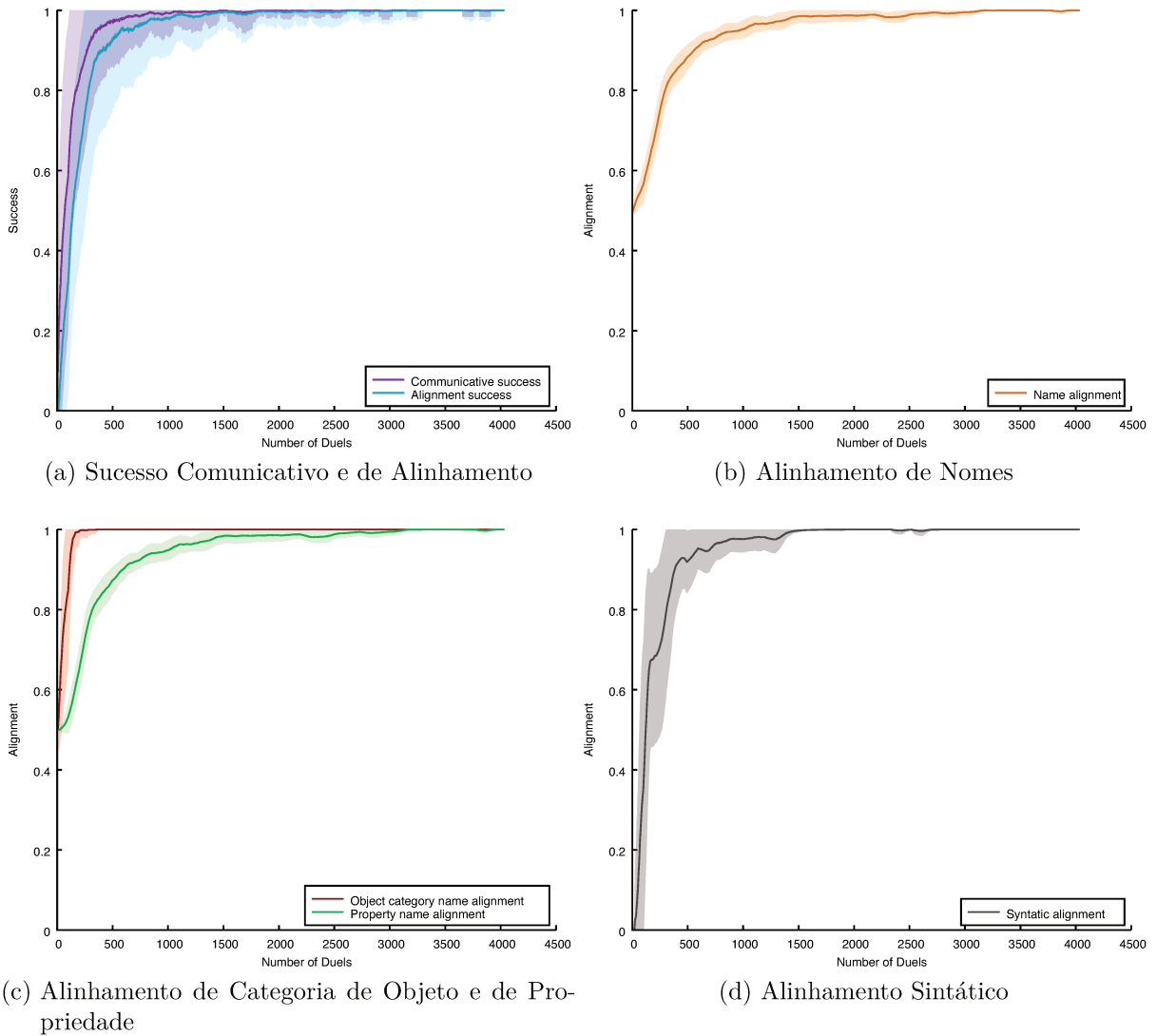


Figura 43 – Cenário T2 com 2 Agentes: Resultado experimental para estratégia de aprendizado gramatical (descrita na seção 6.1.2) e para estratégia de aprendizado sintático (descrita na seção 6.2.1). Parâmetros utilizados na estratégia de aprendizado gramatical: p_a : 1.0, p_{mr} : 0.20, p_{mp} : 0.03, p_{rem} : 0.0005, p_{ne} : 0.8. População de 2 agentes e ringue com 8 figuras geométricas (4 círculos e 4 triângulos). Foram executados aproximadamente 4000 duelos. Em (a) os agentes alcançaram completo sucesso comunicativo e de alinhamento. Em (b) e (c) o alinhamento dos nomes empregados para as categorias de objetos e propriedades também foi completo. Em (d) também houve completo alinhamento sintático.

quentemente, uma breve alteração da evolução do experimento e de seu desvio padrão. De todo modo, esse fato não invalida o resultado, pois, de modo geral, as duas métricas convergiram em 100%. Para tratar essa questão bastaria que uma quantidade maior de duelos fosse executada.

No gráfico (b) podemos perceber que o alinhamento de nomes apresentado é uma média do alinhamento dos nomes empregados para designar as categoria de objetos

e as propriedades. E essa métrica (o alinhamento de nomes) só convergiu completamente quando ambas as categorias também convergiram, como observado no gráfico (c) que permite acompanhar separadamente o alinhamento de nomes das categorias envolvidas no experimento (categorias de objeto e propriedades).

Como pode ser observado no gráfico (c), os nomes das categorias de objetos convergem rapidamente. Isso porque há apenas duas figuras geométricas distintas no experimento (círculo e triângulo). Já os nomes das propriedades levaram mais tempo para convergir, pois o experimento emprega todas as propriedades que permitem discriminar os objetos. Tais propriedades têm quantidade muito superior à das categorias de objetos, portanto a convergência das propriedades requer mais tempo. Consequentemente, mais tempo será necessário para o alinhamento dos dicionários lexicais, influenciando também o sucesso comunicativo e de alinhamento.

Repare que para os alinhamentos de nomes dos gráficos (b) e (c), o experimento iniciou em 50%. Isso ocorreu porque a população é composta por 2 agentes, portanto cada categoria que compõe a sentença possui inicialmente dois nomes competindo para nomeá-la.

Neste experimento também é possível apurar o alinhamento lexical a partir do cálculo da métrica apresentada na seção 3.5.1.1. Entretanto, ela é muito parecida com o sucesso comunicativo, divergindo levemente em contextos onde há a composicionalidade. Como as duas métricas demonstram a mesma coisa, isto é, a preferência dos agentes por um símbolo para representar uma categoria, optamos por apresentar apenas o sucesso comunicativo.

Por fim, o gráfico (d) apresenta a evolução do alinhamento sintático, isto é: a concordância dos agentes sobre a ordem das palavras na formação das sentenças nominais empregadas. Embora a formação das sentenças composicionais seja simples (propriedades + categoria de objeto + propriedades), os agentes devem chegar a um consenso sobre a posição das propriedades. Vale ressaltar que no presente trabalho a posição das propriedades é determinada em função do domínio a que pertencem. Como pode ser observado, no cenário T2 houve um completo alinhamento sintático e, para isso, um pouco mais de 2500 duelos foram necessários. Portanto, a partir dos resultados apurados, pode-se perceber que os agentes alcançaram não só sucesso comunicativo, mas também um pleno alinhamento semântico e sintático.

Para verificar se o *framework* implementado também é capaz de produzir bons resultados em contextos mais desafiadores, o cenário T2 foi repetido em uma população de 6 agentes e as mesmas métricas foram extraídas (sucesso comunicativo e de alinhamento, alinhamento de nomes e alinhamento sintático). Também foram executadas 10 repetições, a partir das quais foram calculadas para cada métrica a média móvel simples de 100 duelos e seu desvio padrão. Os resultados apurados podem ser conferidos na Figura 44.

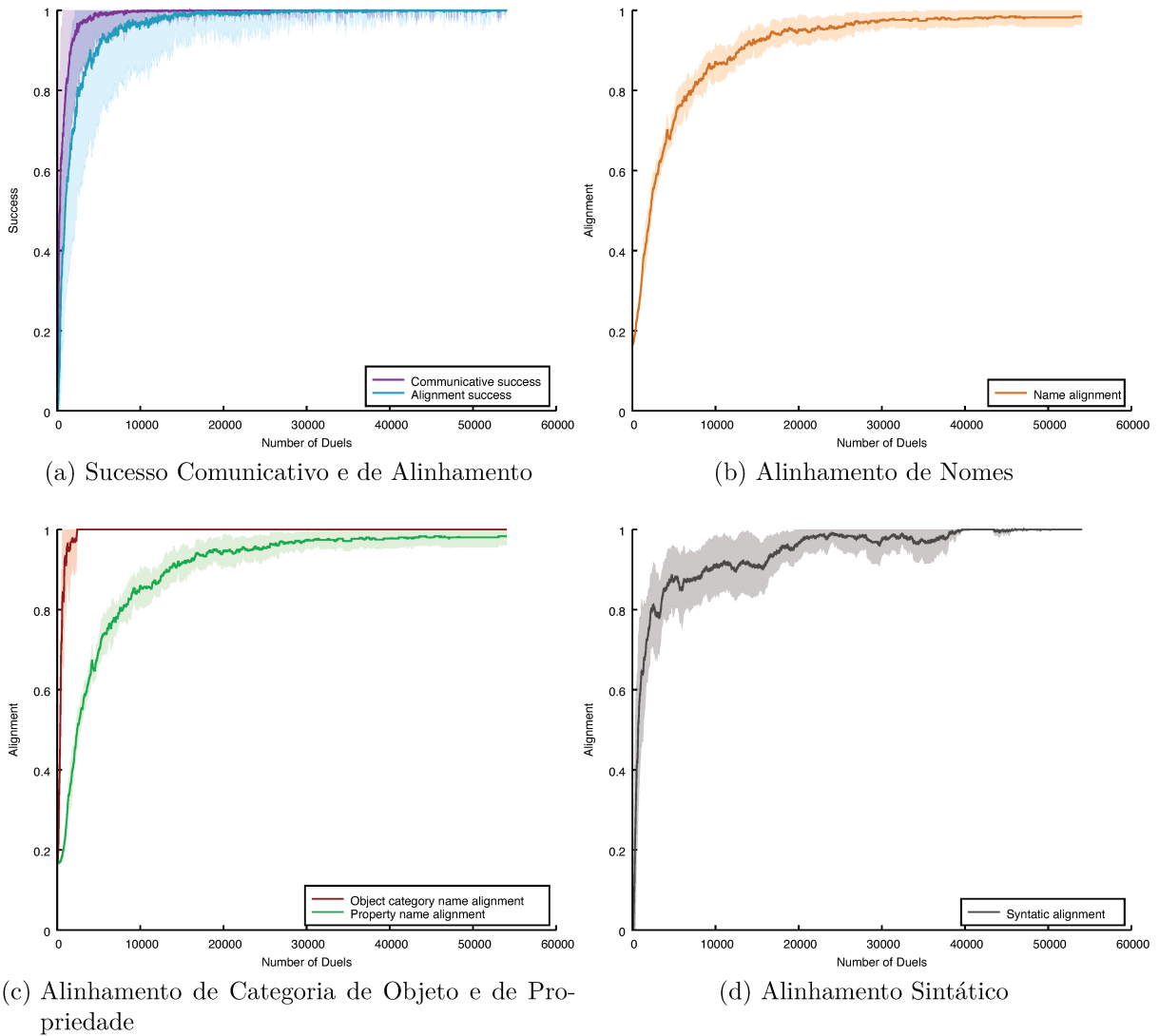


Figura 44 – Cenário T2 com 6 Agentes: Resultado experimental para estratégia de aprendizado gramatical (descrita na seção 6.1.2) e para estratégia de aprendizado sintático (descrita na seção 6.2.1). Parâmetros utilizados na estratégia de aprendizado gramatical: p_a : 1.0, p_{mr} : 0.20, p_{mp} : 0.03, p_{rem} : 0.0005, p_{ne} : 0.8. População de 6 agentes e ringue com 8 figuras geométricas (4 círculos e 4 triângulos). Foram executados aproximadamente 55000 duelos. Em (a) os agentes alcançaram pleno sucesso comunicativo e de alinhamento. Em (b) e (c) o alinhamento dos nomes empregados para as categorias de objetos e propriedades foi alto, porém o alinhamento dos nomes das propriedades não atingiu 100%. Em (d) houve pleno alinhamento sintático.

Analisando os gráficos apresentados na Figura 44, percebe-se nitidamente que houve uma diferença significativa do seu desempenho em relação ao experimento anterior (Figura 43). A primeira informação que ressalta é aquela apresentada pelo gráfico (b), onde a população de agentes não conseguiu alcançar pleno alinhamento de nomes após cerca de 55000 duelos. Como indica o gráfico (c), isso ocorreu porque o alinhamento de nome das propriedades não foi completo. Isso indica que para algumas propriedades, os

agentes não chegaram a um acordo sobre qual nome empregar para nomeá-las.

Entretanto, o gráfico (a) mostra que a população de agentes alcançou pleno sucesso de alinhamento. Como essa métrica envolve o sucesso comunicativo e o alinhamento de nomes, há indicações de que um conjunto pequeno de propriedades não conseguiu alcançar o alinhamento de nomes. Dada a configuração do ambiente, propriedades que raramente eram eleitas para discriminar as figuras geométricas podem ter permanecido sem convergência. Porém, por serem pouco utilizadas, não afetaram significativamente o sucesso de alinhamento. Ao verificar cada uma das 10 repetições realizadas no experimento foi possível constatar que isto ocorreu em 30% das repetições sendo, portanto, uma situação eventual, e não recorrente.

A fim de contrastar o desempenho desse experimento com os demais, as repetições onde a população não alcançou completo alinhamento de nomes foram removidas e novos gráficos foram gerados. A Figura 45 apresenta estes resultados.

Como pode ser observado, o aumento da população influencia no desempenho do experimento, sendo necessários mais duelos para se alcançar completo sucesso comunicativo e de alinhamento. Enquanto com uma população de 2 agentes foram necessários aproximadamente 4000 duelos, para uma população de 6 agentes necessitou-se de pouco mais de 50000 duelos.

Analisando o gráfico (a) da Figura 45 é possível perceber que, assim como no cenário com 2 agentes, o sucesso comunicativo é a métrica que converge primeiro. Isso ocorre porque ela leva em consideração apenas o fato dos agentes conseguirem interpretar a sentença. Já o sucesso de alinhamento leva mais tempo, uma vez que os agentes, além de interpretarem corretamente a sentença, também precisam chegar a um consenso sobre os nomes empregados na sua formação. De todo modo, com quase 30000 duelos já é possível perceber que ambas as métricas atingem 100%.

Novamente é possível perceber oscilações do comportamento médio e isto também se deve ao fato dos dicionários lexicais dos agentes estarem se ajustando para terem um único símbolo para designar uma categoria e uma única categoria para fundamentar um símbolo. Como já afirmado anteriormente, esse comportamento não invalida o resultado, pois analisando o gráfico (b) e (c) é possível verificar que a população tem preferência por um único nome para designar cada figura geométrica e cada propriedade discriminativa.

Comparando o gráfico (b) com o gráfico (c), percebe-se que o alinhamento de nome leva cerca de 50000 duelos para convergir. Essa informação demonstra a complexidade deste experimento, derivada da quantidade alta de propriedades discriminativas. Quanto maior o número de propriedades, maior o grau de incerteza em relação ao significado dos símbolos envolvidos. Como este experimento trata basicamente do alinhamento

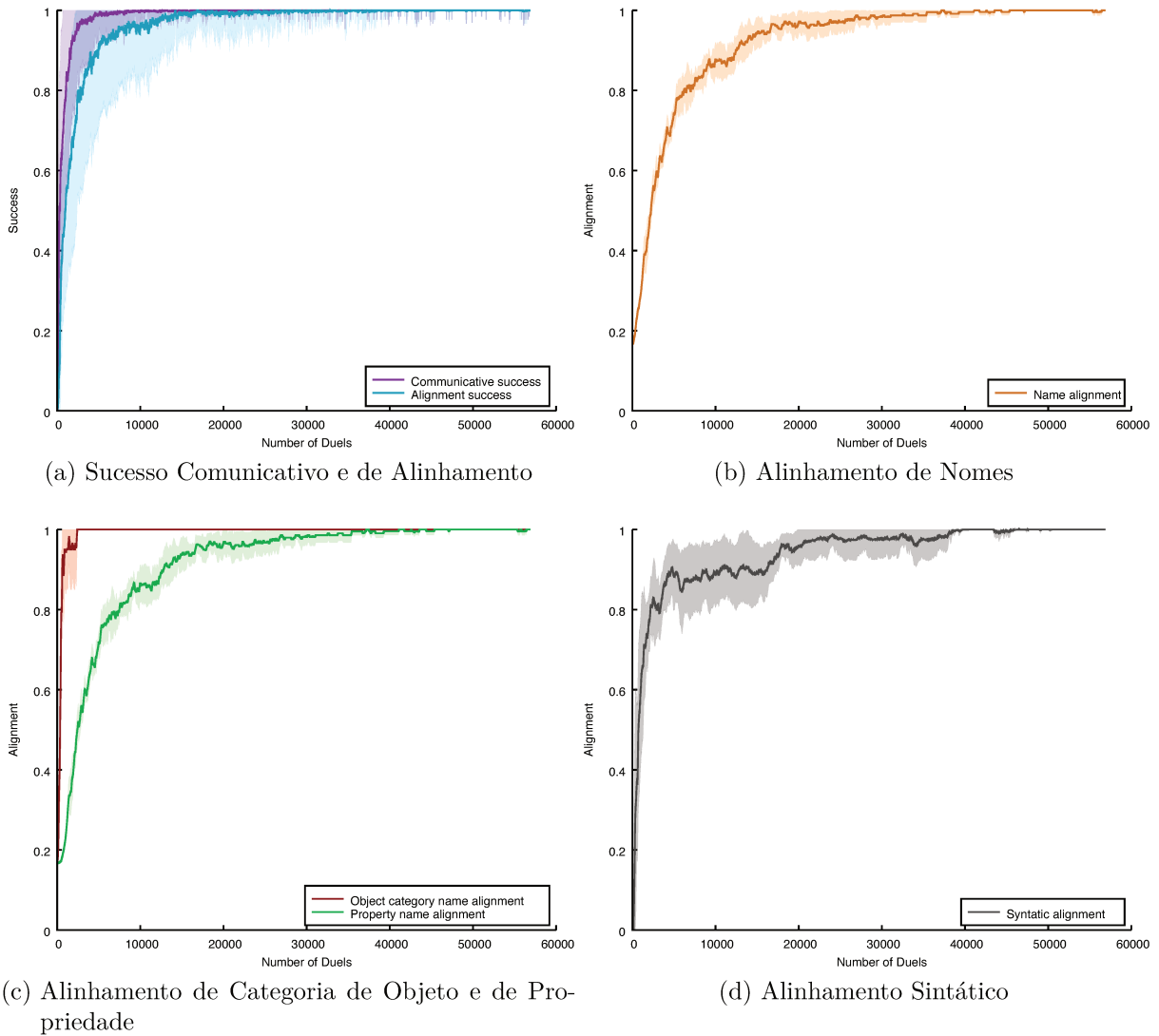


Figura 45 – Cenário T2 com 6 Agentes (completo alinhamento de nomes): Resultado experimental para estratégia de aprendizado gramatical (descrita na seção 6.1.2) e para estratégia de aprendizado sintático (descrita na seção 6.2.1). Parâmetros utilizados na estratégia de aprendizado gramatical: p_a : 1.0, p_{mr} : 0.20, p_{mp} : 0.03, p_{rem} : 0.0005, p_{nc} : 0.8. População de 6 agentes e ringue com 8 figuras geométricas (4 círculos e 4 triângulos). Foram executados aproximadamente 56500 duelos. Em (a) os agentes alcançaram completo sucesso comunicativo e de alinhamento. Em (b) e (c) o alinhamento dos nomes empregados para as categorias de objetos e propriedades foi completo. Em (d) também houve completo alinhamento sintático.

do nome das propriedades, as mesmas são o principal desafio do experimento.

Repare que para os alinhamentos de nomes dos gráficos (b) e (c), o experimento iniciou com 16%. Isso se deve à população ser composta por 6 agentes, conferindo inicialmente, a cada categoria, seis nomes concorrentes.

O alinhamento sintático, por sua vez, só foi completo quando o alinhamento

de nomes das propriedades atingiu valores elevados de convergência. Se compararmos o gráfico (c) com o gráfico (d), podemos observar que o alinhamento sintático só foi completo por volta dos 50000 duelos. Por volta dos 40000 duelos o alinhamento de nome das propriedades já oscilava em torno de 100%.

Apesar do cenário do cenário T2 ser bastante desafiador em razão do número elevado de propriedades envolvidas, ambas as execuções do cenário T2 confirmam que o modelo de representação conceitual e o mecanismo de linguagem propostos conseguiram garantir interoperabilidade semântica e linguística no contexto da composicionalidade. Os agentes da população conseguiram trocar sentenças nominais entre si e interpretá-las corretamente para designar os objetos de seus ambientes. A seguir mais um tipo de experimento é apresentado, no qual é feito o emprego de sentenças composicionais verbais.

7.2.4.4 Cenário T3

O cenário T3 teve como objetivo principal verificar se o mecanismo de simulação mental é capaz de categorizar símbolos composicionais, que neste caso são sentenças verbais empregadas pelos agentes para descrever ações executados no ambiente. Por envolver todas as categorias ontológicas previstas no modelo de representação conceitual este experimento é o mais complexo entre os apresentados no presente trabalho. Assim, pretende-se verificar se o mecanismo de aprendizado também é capaz de garantir a interoperabilidade semântica neste contexto.

Em T3 o ambiente é configurado com um ringue apenas, composto de 6 figuras geométricas: 2 retângulos, 2 quadrados e 2 trapézios. Para compor um evento, são empregadas as categorias ontológicas: objeto conceitual (para categorizar substantivos próprios), categoria de objeto (para categorizar substantivos comuns), propriedades (para categorizar as qualidades dos objetos), *affordance* (para categorizar a ação) e modificadores (para categorizar as maneiras como a ação se desenrolou). Além disso, as *affordances* podem sofrer transformações temporais de passado e futuro. A partir destas categorias, sentenças composicionais verbais podem ser formadas.

No primeiro teste executado foram estabelecidos jogos composicionais em uma população de 2 agentes e, assim como no cenário T2, foram extraídas as seguintes métricas: sucesso comunicativo, sucesso de alinhamento, alinhamento de nome das categorias envolvidas e alinhamento sintático. Foram executadas 10 repetições, a partir das quais calculou-se, para cada métrica mencionada, a média móvel simples de 100 duelos e seu desvio padrão. Os resultados apurados podem ser observados na Figura 46.

O gráfico (a) da Figura 46 mostra que a população de agentes do experimento em questão conseguiu alcançar pleno sucesso comunicativo e de alinhamento. Assim como nos demais experimentos apresentados anteriormente, isto significa que os agentes

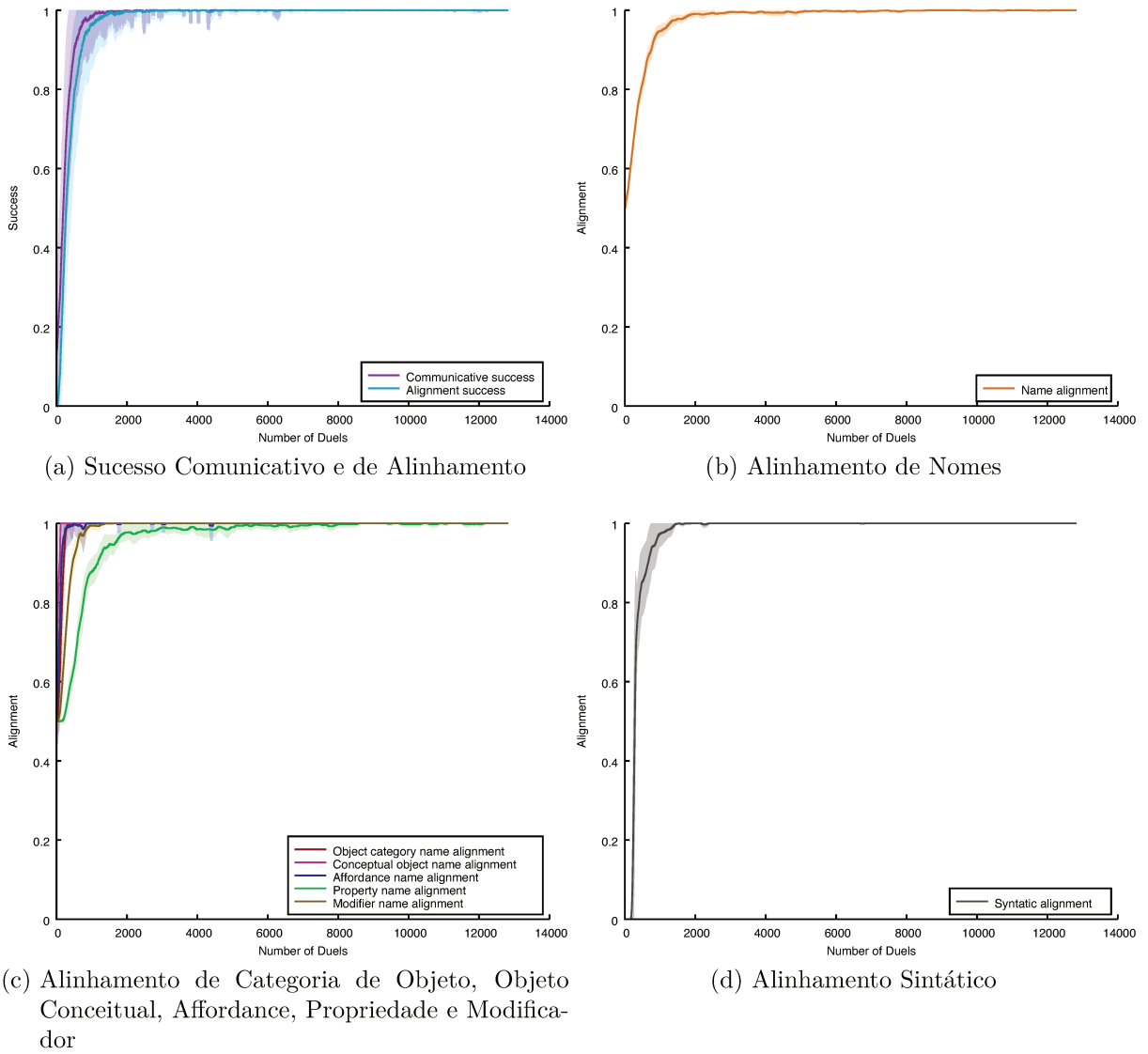


Figura 46 – Cenário T3 com 2 Agentes: Resultado experimental para estratégia de aprendizado gramatical (descrita na seção 6.1.2) e para estratégia de aprendizado sintático (descrita na seção 6.2.1). Parâmetros utilizados na estratégia de aprendizado gramatical: p_a : 1.0, p_{mr} : 0.20, p_{mp} : 0.03, p_{rem} : 0.0005, p_{ne} : 0.8. População de 2 agentes e ringue com 6 figuras geométricas (2 retângulos, 2 quadrados e 2 trapézios). Foram executados aproximadamente 12800 duelos. Em (a) os agentes alcançaram completo sucesso comunicativo e de alinhamento. Em (b) e (c) o alinhamento dos nomes empregados para as categorias de objetos, objetos conceituais, *affordances*, propriedades e modificadores foi completo. Em (d) também houve completo alinhamento sintático.

conseguiram interpretar corretamente as sentenças empregadas para descrever os eventos do ambiente. O que chama atenção neste gráfico é o número de duelos necessários para a completa convergência: tanto no cenário T2 como no cenário T3 com 2 agentes foi necessário cerca de 2000 duelos, e houve menos oscilação no cenário T3. Assim como nos demais experimentos, o sucesso comunicativo convergiu mais rapidamente que o sucesso

de alinhamento.

O gráfico (b) mostra que a população conseguiu chegar a um consenso a respeito dos nomes empregados para compor as sentenças. Mesmo envolvendo 5 categorias ontológicas distintas, para todas elas houve completo alinhamento, como também apresenta o gráfico (c).

Há que se destacar aqui que este experimento envolve categorias mais numerosas, como o caso das propriedades e modificadores, e categorias menos numerosas, como categorias de objetos, objetos conceituais e *affordances*. Por se tratar de um cálculo médio, as categorias menos numerosas convergem mais rapidamente, fazendo com que o alinhamento de nome (gráfico (b)) atinja valores elevados mais rapidamente. Percebe-se que, já após cerca de 3000 duelos, o alinhamento de nome oscilava em torno dos 100% e, assim, era quase completo. Já no gráfico (c), neste mesmo patamar (3000 duelos), o nome das propriedades ainda não havia convergido.

Esta informação indica que o cenário T2 é tão custoso quanto o cenário T3. Para entendermos o motivo, devemos observar que o cenário T2 exercita o alinhamento dos nomes das propriedades, que é o alinhamento mais custoso. Em T3 há também o alinhamento de outras categorias ontológicas, mas isso não afeta muito o tempo de convergência por serem categorias menos custosas. Além disso, o ciclo perceptivo de T2 é mais custoso por atuar sobre 8 objetos, enquanto o ciclo perceptivo de T3 atua sobre apenas 6 objetos.

No gráfico (d) é possível verificar o desempenho da população em relação ao alinhamento sintático, onde há completo alinhamento sintático. Mesmo envolvendo as sentenças mais complexas do nosso estudo, é possível observar que agentes chegaram a um acordo sobre as regras empregadas para estruturar as sentenças trocadas após cerca de 2000 duelos, aproximadamente a mesma quantidade de duelos necessários em T2.

Por fim, percebemos, mais uma vez, que quanto mais rápido se dá o alinhamento de nomes, mais rápido também será o alinhamento das regras sintática. Vejamos a seguir o que ocorre quando o cenário T3 é repetido com uma população de 6 agentes.

A Figura 47 apresenta os resultados apurados do cenário T3 com 6 agentes. O gráfico (a) mostra que mesmo com o aumento da população de agentes foi possível alcançar completo sucesso comunicativo e de alinhamento. Um ponto a ser observado nesse gráfico é a oscilação do comportamento médio da população, que não ocorreu quando se tinha uma população menor. Acredita-se que após a execução de mais duelos esse comportamento seria estabilizado. Comparando esse resultado com aquele apresentado no cenário T2 com 6 agentes (gráfico (a) da Figura 45), com 30000 duelos ambos os experimentos tinham o mesmo comportamento. Isto é, o sucesso comunicativo e de alinhamento dos agentes estava praticamente em 100%.

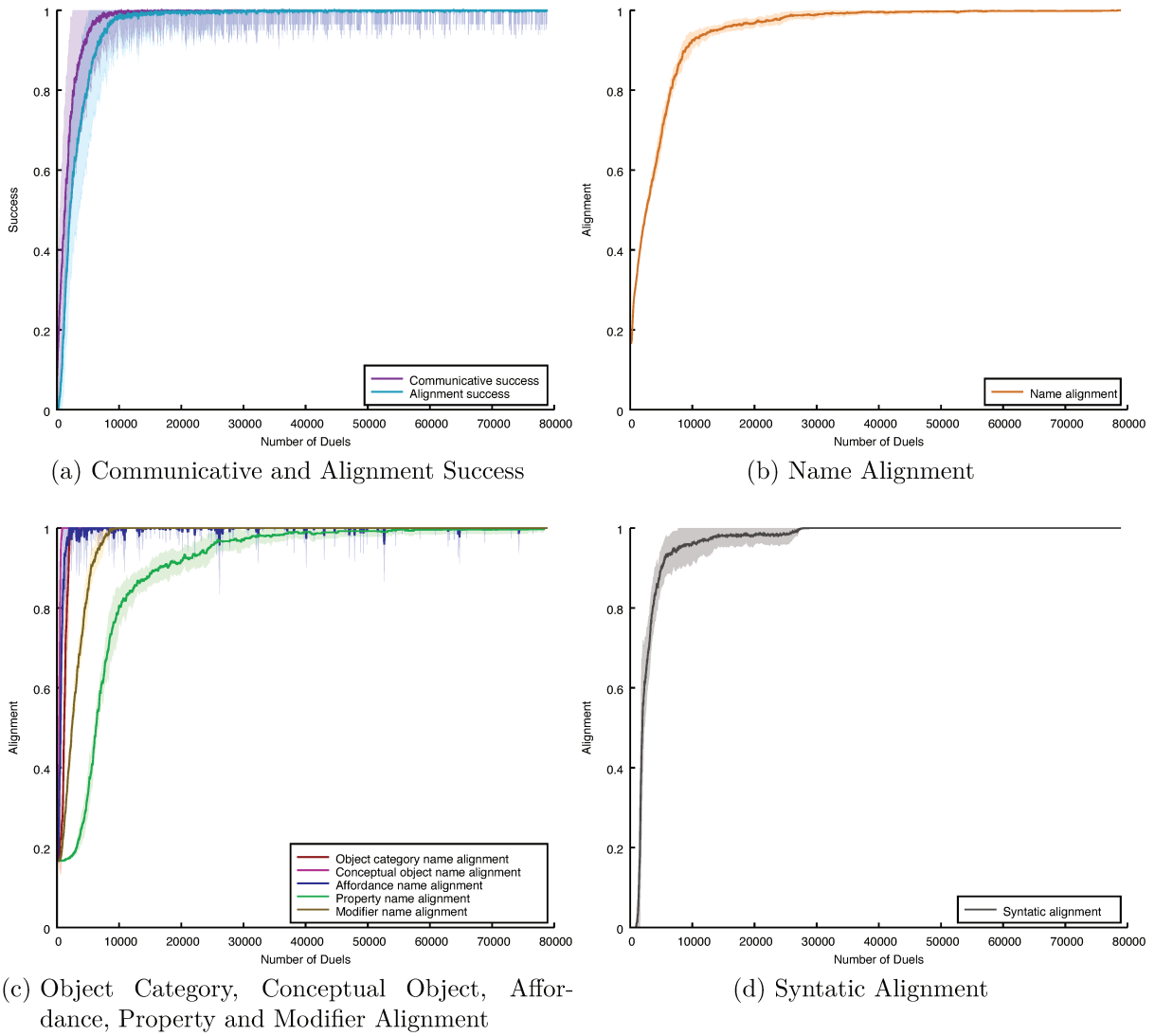


Figura 47 – Cenário T3 com 6 Agentes: Resultado experimental para estratégia de aprendizado gramatical (descrita na seção 6.1.2) e para estratégia de aprendizado sintático (descrita na seção 6.2.1). Parâmetros utilizados na estratégia de aprendizado gramatical: p_a : 1.0, p_{mr} : 0.20, p_{mp} : 0.03, p_{rem} : 0.0005, p_{ne} : 0.8. População de 6 agentes e ringue com 6 figuras geométricas (2 retângulos, 2 quadrados e 2 trapézios). Foram executados aproximadamente 80000 duelos. Em (a) os agentes alcançaram completo sucesso comunicativo e de alinhamento. Em (b) e (c) o alinhamento dos nomes empregados para as categorias de objetos e propriedades foi completo. Em (d) também houve completo alinhamento sintático.

Os gráficos (b) e (c) demonstram o quão alinhada a população está em relação aos nomes empregados para descrever as categorias ontológicas empregadas na formação das sentenças. Como pode ser observado, a população alcançou pleno alinhamento de nomes (gráfico (b)) após cerca de 40000 duelos, similarmente ao cenário T2 com 6 agentes.

Como mostra o gráfico (c), o nome das propriedades e dos modificadores foram os de convergência mais custosa. O alinhamento do nome das propriedades ocorreu após

aproximadamente 60000 duelos, enquanto no cenário T2 com 6 agentes este alinhamento ocorreu por volta dos 50000 duelos.

Por fim, o gráfico (d), apresenta o alinhamento da população de agentes em relação às regras sintáticas utilizadas para estruturar as sentenças empregadas nos duelos. Assim como nos demais experimentos, pode-se verificar que o pleno alinhamento sintático também foi alcançado.

Comparando o gráfico (b) com o gráfico (d), percebe-se que o alinhamento sintático ocorreu a partir de 30000 duelos. Neste ponto o alinhamento de nome da população estava muito próximo dos 100%, indicando mais uma vez que o alinhamento sintático ocorre após o alinhamento de nomes.

Analisando o alinhamento sintático do cenário T2 com 6 agentes, percebe-se que foi a partir dos 40000 duelos que a população alcançou o alinhamento sintático. No caso do experimento atual, ele ocorreu após aproximadamente 30000 duelos.

Portanto, podemos concluir que o *framework* computacional proposto também alcançou bons resultados no contexto do cenário T3. De modo geral, a população de agentes conseguiu empregar sentenças verbais para descrever os eventos do ambiente, interpretá-las corretamente e chegar a um acordo sobre os símbolos empregados e sobre as regras utilizadas para estruturar as sentenças. O bom desempenho do *framework* em T3 é importante, pois trata-se do contexto mais desafiador. Todos os resultados apresentados confirmam a viabilidade do mecanismo proposto.

7.3 Considerações Finais

Neste capítulo foram apresentados os experimentos propostos para validar as funcionalidades do *framework* implementado para o mecanismo de simulação mental.

O primeiro experimento executado empregou um tipo de rede neural, conhecido como mapa auto-organizável evolutivo, como uma ferramenta de aprendizado de máquina capaz de receber dados sensoriais brutos e utilizá-los para segmentar um espaço conceitual. A ferramenta conseguiu, em tempo de execução e sem treinamento prévio, descobrir todas as regiões semânticas do domínio da cor e empregá-las para reificar propriedades. Essas categorias puderam ser facilmente empregadas por uma comunidade de agentes para nomear símbolos em um jogo de nomeação. Os agentes alcançaram sucesso de alinhamento e comunicativo, da mesma forma que no experimento desempenhado por Wellens (2012).

Esse resultado nos leva a crer que a ferramenta atende aos requisitos desejados (ter inspiração biológica, aprender em tempo de execução e sem treinamento prévio) e pode ser agregada ao modelo de representação para particionar o espaço conceitual em

regiões semânticas a partir de dados brutos sensoreados.

Já no segundo experimento procurou-se empregar as categorias ontológicas no processo de emergência de uma linguagem. Para tanto, foram avaliados 3 cenários distintos com populações de 2 e 6 agentes. No primeiro (T1), símbolos isolados foram empregados para descrever objetos do ambiente. No segundo (T2), sentenças nominais foram empregadas para distinguir objetos semelhantes entre si. No terceiro (T3) e último cenário, sentenças verbais foram empregadas para descrever eventos ocorridos no ambiente. Como cada cenário apresenta um grau de dificuldade distinto, os agentes tinham que empregar categorias ontológicas configuradas para fundamentar os símbolos e evoluir um conjunto de regras gramaticais e sintáticas para alcançar sucesso comunicativo e de alinhamento.

Nos cenários onde haviam 2 agentes houve pleno sucesso comunicativo e de alinhamento, além de alinhamento lexical e sintático. Devido à aleatoriedade de cenários como T2 e T3 com 6 agentes, houve algumas execuções que não tiveram comportamento satisfatório. Deve-se investigar o porquê desse ocorrido. De todo modo, esses foram casos isolados e, em sua maioria, as demais repetições tiveram comportamentos satisfatórios. Portanto, a primeira observação que pode ser feita é que o tamanho da população influenciará o desempenho do experimento, especialmente para os cenários de T2 e T3.

De modo geral pode-se afirmar que os agentes alcançaram não apenas sucesso comunicativo, como também sucesso de alinhamento. Isso indica que eles conseguiram gerar as representações icônicas e empregá-las para interpretar os nomes usados para descrever o cenário simulado, como também tinham preferência pelos mesmos nomes. Além do mais, para os cenários T2 e T3, em que foi necessário o emprego de regras sintáticas para formulação das sentenças, houve também pleno sucesso lexical e sintático. Isso significa que os agentes chegaram a um acordo sobre símbolos e suas funções gramaticais, bem como também as regras utilizadas para combiná-los nas sentenças. Vale ressaltar que o aprendizado lexical decorre do fato de associar um símbolo à categoria ontológica correta.

Entre todos os experimentos executados, o T3 com 6 agentes foi o que demandou de mais duelos para alcançar o sucesso comunicativo e de alinhamento. Como era esperado, esse fato ocorreu porque neste experimento há mais agentes e o emprego de sentenças mais complexas, com o maior número de categorias.

Pelo mesmo motivo, também era esperado que o resultado do alinhamento sintático do mesmo cenário tivesse um comportamento inferior ao cenário T2 com 6 agentes. Entretanto, o que se apurou foi o contrário. O cenário T3 é tão custoso quanto o do cenário T2. Podemos justificar esse comportamento no fato das regras sintáticas do experimento T2 envolverem, basicamente, nomes que convergem mais lentamente, que é o caso daqueles que designam as categorias de propriedades. Como consequência, seu

alinhamento sintático também precisará de mais tempo para convergir. Além disso, a configuração do cenário adotado em T2 (4 círculos e 4 triângulos) fez com que seu ciclo perceptivo fosse mais custoso, demandando de mais tempo para ser concluído.

Não há dúvidas de que o processo perceptual seja o ponto do *framework* onde há o maior consumo de recursos computacionais. Talvez em cenários mais desafiadores, ele não seja tão eficiente, como aconteceu com 30% das repetições do cenário T2. A aleatoriedade implícita nesses experimentos merece uma investigação maior.

Ainda poderia ser argumentado que o emprego de jogos de linguagem em populações com 2 e 6 agentes é pouco realístico, devido ao seu tamanho reduzido, quando confrontado com populações humanas que desenvolveram linguagem. Temos consciência de que experimentos com um maior número de agentes seriam desejáveis. Entretanto, nosso experimento basicamente visa aproveitar resultados da literatura que enfatizam que dois sub-sistemas são necessários para o surgimento de linguagem gramatical (ARDILA, 2011; ARDILA, 2012). Nessa linha, nosso *framework* implementa duas funções cognitivas distintas relacionadas à simulação mental (a percepção e a linguagem). Neste caso, dada a complexidade dos algoritmos necessários, é inevitável iniciarmos com experimentos mais simples, que poderão, no futuro, ser estendidos a simulações com um maior número de agentes, quando o poder computacional das máquinas à disposição assim o permitam. Vale ressaltar que o mesmo sucedeu em trabalhos correlatos da literatura, e.g., (VOGT, 2005), onde foram empregadas populações de no máximo 2 agentes para validação das teorias envolvidas.

Além disso, experimentos que simulam jogos de linguagem em populações maiores estão normalmente interessados em estudar outros aspectos da evolução da linguagem e, por esse motivo, acabam simplificando as funções cognitivas envolvidas. Isso pode ser observado, por exemplo, no experimento estabelecido por Wellens (2012), que se ocupou em estudar diferentes estratégias de aprendizado para evolução da linguagem. A arquitetura computacional empregada era bem simples, o que permitiu o uso de populações bem maiores.

Entendemos que nossos experimentos validam nosso *framework* somente em contextos de populações menores. Entretanto, os resultados obtidos são bastante animadores. Em todos os cenários simulados houve a interoperabilidade semântica entre os agentes. Escalar nosso framework para aplicá-lo em cenários que demandam de mais dispositivos interoperantes pode demandar uma arquitetura distribuída e paralela, onde os jogos pudessem ser simulados em paralelo, em diversas máquinas diferentes. Esse esforço, entretanto, foge do escopo do presente trabalho e deixamos aqui a ideia para trabalhos futuros.

8. CONCLUSÃO

Esta tese teve, como objetivo principal, trazer contribuições à área de pesquisa em evolução de linguagem em agentes artificiais, propondo um arcabouço computacional para um mecanismo de simulação mental, capaz de formar e fundamentar sentenças composicionais. Assim sendo, ela aborda um dos problemas filosóficos mais antigos da área de inteligência artificial, o chamado “problema da fundamentação do símbolo”. Entender como símbolos ganham significado, sem depender do projetista do sistema, torna-se relevante para o contexto atual, onde cada vez mais há a necessidade de dispositivos inteligentes autônomos e que apresentem comportamento semelhante ao dos seres vivos. Tal cenário não é diferente no contexto da internet das coisas, especialmente quando se trata de questões envolvendo a interoperabilidade semântica desses sistemas.

Dado a abrangência do tema dessa pesquisa (e sua complexidade), um estudo multidisciplinar foi elaborado. Após a contextualização do problema de pesquisa e sua justificativa na introdução, o capítulo 2 teve como objetivo principal apresentar os trabalhos que compõem o referencial teórico que fundamentaram a construção do mecanismo proposto. A teoria do espaço conceitual de Gärdenfors ganha um papel de destaque por ser um *framework* semântico implementável por meio de um ferramental matemático. Ela ainda é compatível com a teoria semântica das classes de palavras, onde domínios e suas regiões remetem a palavras com diferentes funções gramaticais. Em consonância com os propósitos de Gärdenfors, Barsalou apresenta a teoria do sistema de símbolos perceptuais, destacando o papel dessa função cognitiva (a percepção) para a criação de símbolos fundamentados (símbolos modais).

Embora o foco do trabalho seja o mecanismo de simulação mental, ele é validado no contexto da linguagem, mais especificamente nas aplicações que envolvem jogos de linguagem, o que justifica a importância do capítulo 3, onde apresentamos o conceito dos jogos de linguagem, suas diferentes variações e algumas implementações. O propósito desse capítulo foi preparar o leitor para os conceitos envolvidos em nossa aplicação computacional.

No capítulo 4 buscou-se esboçar um quadro da situação atual da pesquisa nas áreas de: emergência de linguagem gramatical artificial, representação do conhecimento e sistemas perceptuais simbólicos. Estes trabalhos foram avaliados criticamente à luz do referencial teórico pesquisado, a fim de apontar algumas lacunas da área, bem como contextualizar algumas das opções de modelagem feitas.

Na sequência, o processo perceptual do mecanismo de simulação, capaz de gerar representações conceituais, foi apresentado no capítulo 5. A partir dele, diferentes

categorias ontológicas podem ser instanciadas para fundamentar símbolos que remetem a diferentes classes de palavras e inclusive símbolos composicionais. Para complementar, no capítulo 6 foi criado um mecanismo de linguagem que nos deu o suporte para a criação de símbolos e regras gramaticais e sintáticas para aprendizado. Embora esse mecanismo tenha escopo simplificado, ele ainda consegue descrever eventos de contextos reais.

Por fim, no capítulo 7, os resultados dos experimentos realizados para a validação do mecanismo de simulação mental, nos modos de percepção e linguagem, foram apresentados. Os resultados obtidos mostraram-se satisfatórios, tendo em vista o escopo e os objetivos definidos para este trabalho. Ainda neste capítulo, buscou-se também avaliar criticamente algumas limitações existente no *framework*, em seu estado atual, visando caracterizar possibilidades de melhorias, que poderão ser empregadas em trabalhos futuros.

8.1 Principais Contribuições deste Trabalho

O presente trabalho se destaca por sua contribuição na área de emergência de linguagem gramatical artificial e sua fundamentação simbólica. Para tanto, um *framework* computacional foi proposto e desenvolvido. Como resultados desta produção, podemos destacar os seguintes pontos:

1. O trabalho conseguiu juntar, através de um modelo computacional conciso, as teorias dos espaços conceituais e do sistema de símbolos perceptuais, apresentadas respectivamente por Gärdenfors e Barsalou. Nosso trabalho inovou ao apresentar uma implementação computacional para o mecanismo de simulação mental e o emprego de espaços conceituais para derivar diferentes categorias ontológicas, que remetem a diferentes classes gramaticais de palavras. Tal feito parece ser uma contribuição original bastante relevante para a literatura.
2. O mecanismo de simulação mental apresenta um processo perceptual, que utilizando um algoritmo de rastreamento é capaz de detectar eventos ocorridos no ambiente. Esse modo de operação também pode ser considerado como outra contribuição deste trabalho, por possuir uma estrutura independente, que pode facilmente ser utilizada para compor, por exemplo, a estrutura de uma arquitetura cognitiva.
3. Outra contribuição deste trabalho é o mecanismo de linguagem que também foi implementado no mecanismo de simulação mental. As funcionalidades do pacote de linguagem mostram-se relevantes por dispor de uma série de estruturas e métodos que garantem o aprendizado gramatical e sintático. Trata-se de um mecanismo capaz de estruturar sentenças nominais e verbais que descrevem cenários interes-

santes da realidade. Além do mais, ele também pode compor a estrutura de uma arquitetura cognitiva.

4. Como grande parte dos experimentos se deu no âmbito dos jogos de linguagem, foi necessário um escrutínio da literatura pertinente, a partir do qual organizamos um referencial teórico sobre este tema, apresentado no capítulo 3. Em nossa interpretação, entendemos que esse capítulo, por si só, também oferece uma contribuição, pois reúne informações relevantes e organizadas sobre diferentes tipos de jogos. Dificilmente na literatura é possível se localizar um material conciso e que reúna em uma mesma estrutura tais informações.
5. Por fim, ainda podemos citar como contribuição a aplicação computacional desenvolvida, que por meio de uma interface gráfica configurável, permite que diferentes tipos de jogos de linguagem sejam executados. Além disso, a aplicação dispõe de uma série de recursos que facilitam a depuração do cenário sendo rastreado e registra as informações categóricas que são aprendidas e armazenadas na memória. O projeto dessa interface foi desenvolvido com o intuito de uma futura reutilização no contexto de experimentos com jogos de linguagem, funcionando como uma ferramenta de auxílio ao trabalho de quem manuseia uma simulação de evolução de linguagem.

8.2 Perspectivas de Trabalhos Futuros

Esta pesquisa se encerra no momento em que novas e interessantes ideias surgem para aprimorar o mecanismo de simulação desenvolvido. Algumas dessas ideias derivam da possibilidade de melhorias nas capacidades e funcionalidades que foram intencionalmente simplificadas no presente trabalho. Assim, como sugestão de trabalhos futuros, foram relacionadas as seguintes possibilidades:

1. Otimizar o consumo de memória que a aplicação atualmente está exigindo. Sem esta alteração não será possível aplicar o *framework* em contextos que demandam de um número elevado de agentes.
2. Validar o *framework* desenvolvido em cenários reais que envolvam a IoT, como por exemplo, no contexto de uma casa inteligente. Será que neste contexto, o *framework* garantirá que os dispositivos inteligentes alcancem interoperabilidade semântica, assim como foi comprovado no ambiente simulado ?
3. Validar o *framework* em cenários que também envolvam a web semântica.

4. Empregar técnicas de aprendizado de máquina mais robustas, onde todas as categorias ontológicas previstas pelo modelo, em especial as *affordances*, possam ser aprendidas e empregadas na fundamentação simbólica em tempo real.
5. Alterar o modelo de representação para que ele também seja capaz de fundamentar símbolos abstratos, como aqueles que se referem a sentimentos e estados internos não diretamente observáveis, permitindo assim, uma ampliação do *framework*.
6. Alterar o algoritmo de rastreamento para que os objetos possam ser detectados a partir de suas partes, criando, assim, representações hierárquicas entre eles. Talvez, a partir dessa nova funcionalidade, seja possível derivar outras categorias ontológicas para o modelo e, conseqüentemente, novas classes gramaticais.
7. Permitir que objetos de uma mesma categoria e propriedades sejam agrupados em conjunto, possibilitando, assim, a transformação de numeral (plural).
8. Implementar no modelo novas *affordances* para expandir o repertório de eventos que podem ser categorizados pelos agentes.
9. Integrar o módulo de percepção proposto a uma arquitetura cognitiva e verificar se ele pode atuar como tal, gerando representações conceituais de conhecimentos acerca do mundo simulado.
10. Integrar o modelo de representação conceitual a modelos mais sofisticados de gramática, tais como a GCF.
11. Criar cenários onde o mecanismo de linguagem possa ser validado, isoladamente do modelo de representação.
12. No cenário dos experimentos executados, verificar qual seria o comportamento da aplicação caso empregássemos a linguagem de um agente como a dominante em uma população. Será que isso faria com que o tempo de execução dos experimento reduzisse drasticamente?
13. Propor mecanismos mais sofisticados para analisar o processo de causalidade no ambiente. No estado atual do *framework*, assume-se por convenção que os agentes participantes do jogo de linguagem sempre são os sujeitos de uma *affordance*, e os objetos do ambiente são os pacientes da mesma. Essa perspectiva é uma simplificação que foi adotada para a realização dos experimentos. O ideal seria o agente ser capaz de identificar outros agentes e atribuir a eles o papel ontológico de sujeitos das *affordances*. Essa questão da causalidade não é trivial e demandaria todo um estudo a parte.

8.3 Considerações Finais

Diversos trabalhos na literatura têm empregado esforços em entender como símbolos utilizados na linguagem ganham significado, sem depender a priori do arbítrio de um ser humano em sua fundamentação. Especialmente, os diversos trabalhos que focam no aspecto puramente léxico-semântico do desenvolvimento da linguagem (isto é, o uso de palavras isoladas e o seu significado), têm alcançado relativo sucesso. Entretanto, pouco avanço tem sido observado por parte daqueles trabalhos que tentam explorar os aspectos gramaticais da linguagem (uso de símbolos composicionais). E é justamente nesta área que o presente trabalho trouxe uma contribuição.

Revisitando os objetivos de pesquisas propostos na introdução, pode-se concluir que este trabalho trouxe importantes contribuições à pesquisa na área. A partir do módulo de percepção apresentado no capítulo 5 demonstrou-se como agentes inteligentes podem criar categorias ontológicas capazes de descrever diversas características de um ambiente. A instanciamento desse modelo pode ser constatada no capítulo 7, onde ele foi empregado em diferentes implementações de jogos de linguagem.

Para que os jogos pudessem ocorrer, as categorias previstas no mecanismo foram empregadas na fundamentação simbólica das palavras e sentenças utilizadas. Em ambos os cenários simulados, verificou-se que os agentes não apenas conseguiram descrever objetos e eventos ocorridos, como também chegaram a um consenso sobre os símbolos, regras e significados empregados nessa tarefa. Esse resultado nos leva a crer que o modelo proposto resultou na interoperabilidade semântica entre a comunidade de agentes.

Além do processo perceptual, o trabalho propôs um mecanismo de linguagem que conferiu aos agentes as capacidades de criar, estruturar e aprender símbolos durante os duelos estabelecidos entre eles. Dessa forma também, mais um objetivo do trabalho foi alcançado e validado.

A partir dos resultados apurados podemos concluir que o emprego de um tipo de representação intermediária durante o processo de categorização, que segmente a realidade em termos de categorias ontológicas, é capaz de derivar estruturas simbólicas composicionais fundamentadas, validando, portanto, nossa hipótese de trabalho.

Todavia, vale ressaltar que chegar ao final dessa pesquisa só foi possível mediante a superação das dificuldades que a característica multidisciplinar do tema impôs. Mesmo alcançando resultados positivos, entende-se que ela não conseguiu esgotar o tema abordado (a fundamentação de símbolos composicionais) em sua totalidade. Por exemplo, o trabalho em momento algum abordou o emprego de símbolos abstratos.

Essa limitação abre espaço para o seguinte questionamento: será que o modelo proposto é capaz de fundamentar símbolos dessa natureza? Ou ele necessitaria de

uma adaptação, onde recursos adicionais teriam de ser empregados para lidar com esta demanda ? Soluções híbridas, isto é, que empreguem modelos de diferentes vertentes filosóficas (por exemplo, espaços conceituais, soluções simbólicas ou até conexionistas), devem ser empregadas ? Na nossa concepção, parece que esse seja um caminho interessante a se continuar investindo. Mesmo que este trabalho não tenha dado respostas definitivas a todas as questões inerentes ao tema, ele deixa como contribuição seu modelo de representação e seu mecanismo de simulação mental, que poderão ser estendidos em futuras pesquisas, a partir de seu legado.

Referências

- ANDERSON, M. L. Embodied cognition: A field guide. *Artificial intelligence*, Elsevier, v. 149, n. 1, p. 91–130, 2003. Citado 2 vezes nas páginas 21 e 29.
- ANGLUIN, D.; BECERRA-BONACHE, L. Learning meaning before syntax. In: *Grammatical Inference: Algorithms and Applications*. [S.l.]: Springer, 2008. p. 1–14. Citado na página 22.
- ARDILA, A. There are two different language systems in the brain. *Journal of Behavioral and Brain Science*, v. 1, n. 2, p. 23–36, 2011. Citado 3 vezes nas páginas 24, 35 e 162.
- ARDILA, A. Interaction between lexical and grammatical language systems in the brain. *Physics of life reviews*, Elsevier, v. 9, n. 2, p. 198–214, 2012. Citado 2 vezes nas páginas 22 e 162.
- ATZORI, L.; IERA, A.; MORABITO, G. The internet of things: A survey. *Computer networks*, Elsevier, v. 54, n. 15, p. 2787–2805, 2010. Citado na página 20.
- BAILLIE, J.-C.; GANASCIA, J.-G. Action categorization from video sequences. In: IOS PRESS. *Proceedings of the 14th European Conference on Artificial Intelligence*. [S.l.], 2000. p. 643–647. Citado 3 vezes nas páginas 69, 78 e 180.
- BALKENIUS, C.; GÄRDENFORS, P.; HALL, L. The origin of symbols in the brain. In: *Proceedings of the 3rd International Evolution of Language Conference, Ecole Nationale Supérieure des Telecommunications*. [S.l.: s.n.], 2000. p. 13–17. Citado na página 31.
- BARSALOU, L. W. Perceptual symbol systems. *Behavioral and brain sciences*, Cambridge Univ Press, v. 22, n. 04, p. 577–660, 1999. Citado 11 vezes nas páginas 23, 27, 32, 36, 37, 38, 46, 70, 78, 99 e 184.
- BARSALOU, L. W. Simulation, situated conceptualization, and prediction. *Philosophical Transactions of the Royal Society B: Biological Sciences*, The Royal Society, v. 364, n. 1521, p. 1281–1289, 2009. Citado na página 199.
- BATALI, J. Computational simulations of the emergence of grammar. *Approaches to the evolution of language: Social and cognitive bases*, Cambridge University Press Cambridge, v. 405, p. 426, 1998. Citado na página 22.
- BERGEN, B.; CHANG, N. Embodied construction grammar. In: *The Oxford handbook of construction grammar*. [S.l.: s.n.], 2013. Citado 4 vezes nas páginas 74, 75, 78 e 190.
- BICKERTON, D. *Adam's tongue: how humans made language, how language made humans*. [S.l.]: Macmillan, 2009. Citado na página 100.
- BOAS, H. C.; SAG, I. A. *Sign-based construction grammar*. [S.l.]: CSLI Publications/Center for the Study of Language and Information, 2012. Citado na página 74.
- BORENSZTAJN, G.; TREHAN, V. Luc steels, the talking heads experiment and cognitive philosophy. *A tutorial*, 2006. Citado 3 vezes nas páginas , 51 e 52.

- BRISCOE, E. The acquisition of grammar in an evolving population of language agents. *Electron. Trans. Artif. Intell.*, Citeseer, v. 3, n. B, p. 47–77, 1999. Citado na página 22.
- BROOKS, R. A. Elephants don't play chess. *Robotics and autonomous systems*, Elsevier, v. 6, n. 1, p. 3–15, 1990. Citado 2 vezes nas páginas 21 e 31.
- CHALMERS, D. J.; FRENCH, R. M.; HOFSTADTER, D. R. High-level perception, representation, and analogy: A critique of artificial intelligence methodology. *Journal of Experimental & Theoretical Artificial Intelligence*, Taylor & Francis, v. 4, n. 3, p. 185–211, 1992. Citado na página 36.
- CHANDRASEKARAN, B.; JOSEPHSON, J. R.; BENJAMINS, V. R. What are ontologies, and why do we need them? *IEEE Intelligent systems*, IEEE, n. 1, p. 20–26, 1999. Citado na página 30.
- CHANG, N. C.-L. *Constructing grammar: A computational model of the emergence of early constructions*. [S.l.]: ProQuest, 2008. Citado na página 22.
- CLANCEY, W. J. *Situated cognition: On human knowledge and computer representations*. [S.l.]: Cambridge University Press, 1997. Citado 2 vezes nas páginas 21 e 31.
- CLARK, A. *Being there: Putting brain, body, and world together again*. [S.l.]: MIT press, 1998. Citado na página 22.
- CROFT, W. *Radical construction grammar: Syntactic theory in typological perspective*. [S.l.]: Oxford University Press on Demand, 2001. Citado na página 74.
- DAVIS, R.; SHROBE, H.; SZOLOVITS, P. What is a knowledge representation? *AI magazine*, v. 14, n. 1, p. 17–17, 1993. Citado na página 29.
- DENG, D.; KASABOV, N. Esom: An algorithm to evolve self-organizing maps from on-line data streams. In: IEEE COMPUTER SOCIETY. *Neural Networks, IEEE-INNS-ENNS International Joint Conference on*. [S.l.], 2000. v. 6, p. 6003–6003. Citado 4 vezes nas páginas , 122, 123 e 124.
- DIETZE, S.; DOMINGUE, J. Exploiting conceptual spaces for ontology integration. 2008. Citado 2 vezes nas páginas 21 e 30.
- DOMINEY, P. F.; HOEN, M.; INUI, T. A neurolinguistic model of grammatical construction processing. *Journal of Cognitive Neuroscience*, MIT Press, v. 18, n. 12, p. 2088–2107, 2006. Citado 3 vezes nas páginas 73, 78 e 188.
- FERRARI, L. *Introdução à linguística cognitiva*. [S.l.]: Editora Contexto, 2011. Citado na página 73.
- FRANK, A. U. Ontology for spatio-temporal databases. In: *Spatio-Temporal Databases*. [S.l.]: Springer, 2003. p. 9–77. Citado na página 30.
- FRANK, S. L.; VIGLIOCCO, G. Sentence comprehension as mental simulation: an information-theoretic perspective. *Information, Molecular Diversity Preservation International*, v. 2, n. 4, p. 672–696, 2011. Citado 3 vezes nas páginas 70, 78 e 184.

- FRANKLIN, S.; MADL, T.; STRAIN, S.; FAGHIHI, U.; DONG, D.; KUGELE, S.; SNAIDER, J.; AGRAWAL, P.; CHEN, S. A lida cognitive model tutorial. *Biologically Inspired Cognitive Architectures*, Elsevier, v. 16, p. 105–130, 2016. Citado na página 68.
- GÄRDENFORS, P. *Conceptual spaces: The geometry of thought*. [S.l.]: MIT press, 2004. Citado 6 vezes nas páginas 38, 39, 40, 41, 121 e 125.
- GÄRDENFORS, P. *The geometry of meaning: Semantics based on conceptual spaces*. [S.l.]: MIT Press, 2014. Citado 5 vezes nas páginas 23, 38, 39, 46 e 70.
- GÄRDENFORS, P. A semantic theory of word classes. *Croatian Journal of Philosophy*, v. 14, n. 2, p. 179–194, 2014. Citado 11 vezes nas páginas 41, 42, 43, 44, 45, 46, 84, 85, 199, 201 e 202.
- GÄRDENFORS, P.; WARGLIEN, M. Using conceptual spaces to model actions and events. *Journal of Semantics*, Oxford Univ Press, p. ffs007, 2012. Citado na página 41.
- GIBSON, J. J. *The ecological approach to visual perception*. [S.l.]: Psychology Press, 1986. Citado 4 vezes nas páginas 25, 56, 85 e 201.
- GLENBERG, A. M.; GALLESE, V. Action-based language: a theory of language acquisition, comprehension, and production. *Cortex*, Elsevier, v. 48, n. 7, p. 905–922, 2012. Citado 3 vezes nas páginas 72, 78 e 187.
- GLENBERG, A. M.; ROBERTSON, D. A. Symbol grounding and meaning: A comparison of high-dimensional and embodied theories of meaning. *Journal of memory and language*, Elsevier, v. 43, n. 3, p. 379–401, 2000. Citado 2 vezes nas páginas 21 e 31.
- GRUBER, T. R. A translation approach to portable ontology specifications. *Knowledge acquisition*, Elsevier, v. 5, n. 2, p. 199–220, 1993. Citado 2 vezes nas páginas 21 e 30.
- GUARINO, N.; OBERLE, D.; STAAB, S. What is an ontology? In: *Handbook on ontologies*. [S.l.]: Springer, 2009. p. 1–17. Citado na página 199.
- GUDWIN, R. R. Modelos intermediários de representação mental: A interpretação objetual do ambiente como requisito para aquisição de linguagem gramatical em sistemas artificiais. In: _____. *Temas em Ciências Cognitivas e Representação Mental*. [S.l.]: Synopsys, 2012. cap. 12, p. 279–300. Citado na página 24.
- HARNAD, S. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, Elsevier, v. 42, n. 1, p. 335–346, 1990. Citado 3 vezes nas páginas 30, 31 e 32.
- HUTCHINS, E.; HAZLEHURST, B. How to invent a lexicon: the development of shared symbols in interaction. *Artificial societies: The computer simulation of social life*, London: UCL Press, p. 157–189, 1995. Citado 2 vezes nas páginas 22 e 77.
- JAIN, A. K.; MURTY, M. N.; FLYNN, P. J. Data clustering: a review. *ACM computing surveys (CSUR)*, Acm, v. 31, n. 3, p. 264–323, 1999. Citado na página 121.
- JOHNSON-LAIRD, P. N. *Mental models: Towards a cognitive science of language, inference, and consciousness*. [S.l.]: Harvard University Press, 1983. v. 6. Citado na página 21.

- JOHNSON-LAIRD, P. N.; BYRNE, R. M. Conditionals: a theory of meaning, pragmatics, and inference. *Psychological review*, American Psychological Association, v. 109, n. 4, p. 646, 2002. Citado na página 21.
- JULIER, S. J.; UHLMANN, J. K. New extension of the kalman filter to nonlinear systems. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. *Signal processing, sensor fusion, and target recognition VI*. [S.l.], 1997. v. 3068, p. 182–193. Citado 2 vezes nas páginas 86 e 204.
- JUNG, D.; ZELINSKY, A. Grounded symbolic communication between heterogeneous cooperating robots. *Autonomous Robots*, Springer, v. 8, n. 3, p. 269–292, 2000. Citado 2 vezes nas páginas 22 e 77.
- KHAYI, N. A.; FRANKLIN, S. Initiating language in lida: Learning the meaning of vervet alarm calls. *Biologically Inspired Cognitive Architectures*, Elsevier, v. 23, p. 7–18, 2018. Citado 3 vezes nas páginas 68, 69 e 179.
- KIRBY, S. Syntax without natural selection: How compositionality emerges from vocabulary in a population of learners. *The evolutionary emergence of language: Social function and the origins of linguistic form*, v. 302, p. 323, 2000. Citado 2 vezes nas páginas 22 e 72.
- KOHONEN, T. Self-organized formation of topologically correct feature maps. *Biological cybernetics*, Springer, v. 43, n. 1, p. 59–69, 1982. Citado na página 122.
- KOTSERUBA, I.; TSOTSOS, J. K. 40 years of cognitive architectures: core cognitive abilities and practical applications. *Artificial Intelligence Review*, Springer, p. 1–78, 2018. Citado na página 67.
- LAKOFF, G.; JOHNSON, M. *Philosophy in the flesh: The embodied mind and its challenge to western thought*. [S.l.]: Basic books, 1999. Citado na página 22.
- LALLEE, S.; MADDEN, C.; HOEN, M.; DOMINEY, P. F. Linking language with embodied and teleological representations of action for humanoid cognition. *Frontiers in neurorobotics*, Frontiers Research Foundation, v. 4, 2010. Citado 3 vezes nas páginas 74, 78 e 189.
- LANGACKER, R. W. Cognitive (construction) grammar. *Cognitive Linguistics*, Walter de Gruyter GmbH & Co. KG, v. 20, n. 1, p. 167–176, 2009. Citado na página 74.
- LAVIOLA, A.; LAGE, L.; MARÇÃO, N.; TAVARES, T.; ALMEIDA, V.; MATOS, E.; TORRENT, T. The brazilian portuguese constructicon: Modeling constructional inheritance, frame evocation and constraints in framenet brasil. In: *2017 AAAI Spring Symposium Series*. [S.l.: s.n.], 2017. Citado na página 75.
- LIETO, A. A computational framework for concept representation in cognitive systems and architectures: Concepts as heterogeneous proxytypes. *Procedia Computer Science*, Elsevier, v. 41, p. 6–14, 2014. Citado 2 vezes nas páginas 69 e 179.
- LIETO, A.; RADICIONI, D. P.; RHO, V. Dual peccs: a cognitive system for conceptual representation and categorization. *Journal of Experimental & Theoretical Artificial Intelligence*, Taylor & Francis, v. 29, n. 2, p. 433–452, 2017. Citado 3 vezes nas páginas 69, 78 e 180.

- LOULA, A. *Comunicação Simbólica entre Criaturas Artificiais: um experimento em vida artificial*. Dissertação (Mestrado) — Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, 2004. Citado 4 vezes nas páginas 22, 68, 69 e 78.
- LOULA, A.; GUDWIN, R.; EL-HANI, C. N.; QUEIROZ, J. Emergence of self-organized symbol-based communication in artificial creatures. *Cognitive Systems Research*, v. 11, n. 2, p. 131 – 147, 2010. ISSN 1389-0417. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1389041708000703>>. Citado na página 68.
- LOULA, A.; GUDWIN, R.; QUEIROZ, J.; RIBEIRO, S.; ARAÚJO, I. de. Synthetic approach of symbolic creatures. *SEED Journal—Semiotics, Evolution, Energy, and Development*, v. 3, n. 3, p. 125–133, 2003. Citado na página 68.
- LOULA, A.; GUDWIN, R.; RIBEIRO, S.; QUEIROZ, J. On building meaning: a biologically-inspired experiment on symbol-based communication. In: *Brain inspired cognitive systems 2008*. [S.l.]: Springer, 2010. p. 77–93. Citado na página 68.
- MADDEN, C.; HOEN, M.; DOMINEY, P. F. A cognitive neuroscience perspective on embodied language for human–robot cooperation. *Brain and language*, Elsevier, v. 112, n. 3, p. 180–188, 2010. Citado 5 vezes nas páginas 73, 74, 78, 188 e 189.
- MAJEWSKI, M.; ZURADA, J. M. Sentence recognition using artificial neural networks. *Knowledge-Based Systems*, Elsevier, v. 21, n. 7, p. 629–635, 2008. Citado na página 22.
- MARTÍN, S.; PAYAM, B.; FRANCOIS, C.; PHILIPPE, C.; OVIDIU, V.; PETER, F. *IoT Semantic Interoperability Research Challenges, Best Practices, Recommendations and Next Steps*. [S.l.], 2015. Citado na página 20.
- MATURANA, H. R.; VARELA, F. J. *A árvore do conhecimento: as bases biológicas da compreensão humana; The tree of knowledge: the biological basis of human understanding*. [S.l.]: Palas Athena, 2001. Citado na página 21.
- MELARA, R. D. The concept of perceptual similarity: From psychophysics to cognitive psychology. *Psychophysical approaches to cognition*, Elsevier Science Publishers BV, p. 303–388, 1992. Citado na página 39.
- MUNKRES, J. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, SIAM, v. 5, n. 1, p. 32–38, 1957. Citado na página 97.
- NARAYANAN, S. Moving right along: A computational model of metaphoric reasoning about events. In: *AAAI/IAAI*. [S.l.: s.n.], 1999. p. 121–127. Citado 4 vezes nas páginas 70, 75, 78 e 182.
- NOË, A. *Action in perception*. [S.l.]: MIT Press, 2004. Citado 2 vezes nas páginas 22 e 36.
- OLIPHANT, M.; BATALI, J. Learning and the emergence of coordinated communication. *Center for research on language newsletter*, Citeseer, v. 11, n. 1, p. 1–46, 1997. Citado 2 vezes nas páginas 22 e 77.

- OUDEYER, P.-y. Self-organization of a lexicon in a structured society of agents. In: *Advances in Artificial Life*. [S.l.]: Springer, 1999. p. 725–729. Citado 2 vezes nas páginas 22 e 77.
- PEIRCE, C. *Collected Papers of Charles Sanders Peirce: Edited by Charles Hartshorne and Paul Weiss: Volumes I–VI*. [S.l.]: Harvard University Press, 1958. Citado na página 33.
- PEZZULO, G.; BARSALOU, L. W.; CANGELOSI, A.; FISCHER, M. H.; MCRAE, K.; SPIVEY, M. J. The mechanics of embodiment: a dialog on embodiment and computational modeling. *Frontiers in psychology*, Frontiers Media SA, v. 2, 2011. Citado 2 vezes nas páginas 21 e 31.
- POLCEANU, M.; BUCHE, C. Computational mental simulation: A review. *Computer Animation and Virtual Worlds*, Wiley Online Library, v. 28, n. 5, p. e1732, 2017. Citado na página 71.
- ROCHA, Y. G.; KUC, T.-Y. Mental simulation for autonomous learning and planning based on triplet ontological semantic model. 2019. Citado na página 71.
- ROY, D. Semiotic schemas: A framework for grounding language in action and perception. *Artificial Intelligence*, Elsevier, v. 167, n. 1, p. 170–205, 2005. Citado 3 vezes nas páginas 70, 78 e 183.
- ROY, D.; REITER, E. Connecting language to the world. *Artificial Intelligence*, Elsevier, v. 167, n. 1, p. 1–12, 2005. Citado 4 vezes nas páginas 21, 31, 70 e 78.
- ROY, D. K. Learning visually grounded words and syntax for a scene description task. *Computer Speech & Language*, Elsevier, v. 16, n. 3, p. 353–385, 2002. Citado 2 vezes nas páginas 22 e 78.
- RUMELHART, D. E.; MCCLELLAND, J. L. *Parallel distributed processing: Explorations in the microstructure of cognition: Foundations (Parallel distributed processing)*. [S.l.]: MIT Press, August, 1986. Citado na página 32.
- SAMSONOVICH, A. V. Toward a unified catalog of implemented cognitive architectures. *BICA*, v. 221, n. 2010, p. 195–244, 2010. Citado na página 67.
- SANTAELLA, L. *A teoria geral dos signos: como as linguagens significam as coisas*. [S.l.]: Pioneira, 2000. Citado na página 34.
- SEARLE, J. R. *et al.* Minds, brains, and programs. *Behavioral and brain sciences*, Cambridge Univ Press, v. 3, n. 3, p. 417–457, 1980. Citado na página 31.
- SIERRA-SANTIBÁÑEZ, J. An agent-based model studying the acquisition of a language system of logical constructions. In: *Twenty-Eighth AAAI Conference on Artificial Intelligence*. [S.l.: s.n.], 2014. Citado 3 vezes nas páginas 72, 78 e 186.
- SILVA, I. N. da; SPATTI, D. H.; FLAUZINO, R. A. *Redes Neurais artificiais para engenharia e ciências aplicadas curso prático*. [S.l.]: Artliber, 2010. Citado 2 vezes nas páginas 121 e 122.

- SISKIND, J. M. Grounding the lexical semantics of verbs in visual perception using force dynamics and event logic. *Journal of artificial intelligence research*, v. 15, p. 31–90, 2001. Citado 3 vezes nas páginas 70, 78 e 181.
- SMITH, L. B. dimensions in development. *Similarity and analogical reasoning*, Cambridge University Press, p. 146, 1989. Citado na página 39.
- SOWA, J. F. *et al.* *Knowledge representation: logical, philosophical, and computational foundations*. [S.l.]: Brooks/Cole Pacific Grove, CA, 2000. v. 13. Citado na página 29.
- SPRANGER, M. Incremental recruitment language—a formalism for evolutionary semantics. In: WORLD SCIENTIFIC. *Evolution of Language: Proceedings of the 10th International Conference (EVLANG10)*. [S.l.], 2014. p. 523–524. Citado na página 58.
- SPRANGER, M.; STEELS, L. Emergent functional grammar for space. *Experiments in Cultural Language Evolution*, John Benjamins Amsterdam, v. 3, p. 207–232, 2012. Citado 2 vezes nas páginas 76 e 191.
- STEELS, L. The talking heads experiment. volume 1. words and meanings. *Special pre-print for the LABORATORIUM Antwerpen*, 1999. Citado 2 vezes nas páginas 22 e 77.
- STEELS, L. Evolving grounded communication for robots. *Trends in cognitive sciences*, Elsevier, v. 7, n. 7, p. 308–312, 2003. Citado na página 22.
- STEELS, L. The symbol grounding problem has been solved. so what’s next. *Symbols and embodiment: Debates on meaning and cognition*, Oxford University Press USA, p. 223–244, 2008. Citado na página 33.
- STEELS, L. *The Talking Heads experiment: Origins of words and meanings*. [S.l.]: Language Science Press, 2015. v. 1. Citado 4 vezes nas páginas 49, 50, 51 e 54.
- STEELS, L. Basics of fluid construction grammar. *Constructions and frames*, v. 9, n. 2, p. 178–255, 2017. Citado 4 vezes nas páginas 58, 74, 75 e 191.
- STEELS, L.; SPRANGER, M.; TRIJP, R. V.; HÖFER, S.; HILD, M. Emergent action language on real robots. In: *Language grounding in robots*. [S.l.]: Springer, 2012. p. 255–276. Citado 6 vezes nas páginas 56, 57, 58, 59, 76 e 192.
- TADDEO, M.; FLORIDI, L. Solving the symbol grounding problem: a critical review of fifteen years of research. *Journal of Experimental & Theoretical Artificial Intelligence*, Taylor & Francis, v. 17, n. 4, p. 419–445, 2005. Citado na página 33.
- THOMPSON, E.; VARELA, F. J. Radical embodiment: neural dynamics and consciousness. *Trends in cognitive sciences*, Elsevier, v. 5, n. 10, p. 418–425, 2001. Citado 2 vezes nas páginas 21 e 32.
- TRIJP, R. V. The evolution of case systems for marking event structure. *Experiments in cultural language evolution*, John Benjamins Publishing, v. 3, p. 169–205, 2012. Citado 3 vezes nas páginas 59, 76 e 193.
- VOGT, P. The emergence of compositional structures in perceptually grounded language games. *Artificial intelligence*, Elsevier, v. 167, n. 1, p. 206–242, 2005. Citado 4 vezes nas páginas 72, 79, 162 e 185.

VOGT, P.; COUMANS, H. Exploring the impact of contextual input on the evolution of word-meaning. In: MIT PRESS. *From Animals to Animats 7: Proceedings of the Seventh International Conference on Simulation of Adaptive Behavior*. [S.l.], 2002. v. 7, p. 405. Citado 2 vezes nas páginas 22 e 78.

WELLENS, P. Adaptive strategies in the emergence of lexical systems. In: SPRINGER VERLAG. *Proceedings of the 21st Belgian-Dutch Conference on Machine Learning*. [S.l.], 2012. v. 20, p. 76–88. Citado 12 vezes nas páginas , 54, 55, 59, 60, 62, 104, 127, 129, 130, 160 e 162.

WITTGENSTEIN, L. *Philosophical investigations*. [S.l.]: John Wiley & Sons, 2009. Citado na página 48.

XU, R.; WUNSCH, D. *et al.* Survey of clustering algorithms. *Neural Networks, IEEE Transactions on*, Ieee, v. 16, n. 3, p. 645–678, 2005. Citado na página 121.

ZIEMKE, T. Rethinking grounding. In: *Understanding representation in the cognitive sciences*. [S.l.]: Springer, 2000. p. 177–190. Citado 2 vezes nas páginas 21 e 31.

ZIEMKE, T. Are robots embodied. In: CITESEER. *First international workshop on epigenetic robotics Modeling Cognitive Development in Robotic Systems*. [S.l.], 2001. v. 85. Citado 2 vezes nas páginas 21 e 31.

ZWAAN, R. A. Situation models, mental simulations, and abstract concepts in discourse comprehension. *Psychonomic bulletin & review*, Springer, v. 23, n. 4, p. 1028–1034, 2016. Citado na página 79.

ANEXOS

A. SUMÁRIO DE TRABALHOS CORRELATOS

Tabela 4 – Trabalhos correlatos apresentados no Capítulo 4.

Contexto de Aplicação do Trabalho		Autor	Breve descrição	Pontos Fortes	Críticas
Frameworks Computacionais Gerais	Arquiteturas Cognitivas	Khayi e Franklin (2018)	Simular um sistema de comunicação animal biológico, por meio do qual macacos vervetes aprendem o significado de sons de alarme.	- O mecanismo perceptual da arquitetura está apto para tratar a questão da fundamentação do símbolo; - Das diversas arquiteturas cognitivas apresentadas na literatura, ela é a única que emprega mecanismo de emergência da linguagem com a preocupação de manipular símbolos fundamentados.	- Os símbolos manipulados são isolados, uma questão que já foi tratado pela literatura. Portanto, não menciona a questão da composicionalidade.
Frameworks Computacionais Gerais	Representação Conceitual	Lieto (2014)	Framework de representação de conceitos para sistemas artificiais inteligentes e arquiteturas cognitivas.	- Fornece uma nova hipótese para o problema da representação do conhecimento nas ciências cognitivas, trazendo uma abordagem heterogênea para representação dos conceitos.	- Não se ocupada da implementação da proposta

Contexto de Aplicação do Trabalho		Autor	Breve descrição	Pontos Fortes	Críticas
		Lieto <i>et al.</i> (2017)	Para validar as ideias do framework proposto anteriormente, desenvolveram a Dual-PECC (<i>Prototypes and Exemplars-based Conceptual Categorization System</i>), um sistema de representação do conhecimento e raciocínio.	- Manipula ao mesmo tempo conceitos simbólicos e aqueles oriundos de espaços conceituais; - Conseguiu integrar o sistema em duas arquiteturas bem conhecidas na literatura (ACT-R e Clarity).	- Não aborda como os conceitos podem ser aprendidos. Portanto, ainda há uma dependência do projetista do sistema; - Conceitos evocam apenas símbolos isolados que designam coisas e entidades do mundo.
Frameworks Computacionais Gerais	Sistema Perceptual e Motor	Baillie e Ganascia (2000)	Apresenta uma técnica de aprendizado não supervisionada para extrair informações relevantes (descrições de ações) de sequências de vídeo.	- Método de aprendizado não supervisionado; - Fornece uma representação semântica para ação.	- Reconhece a ação com base no perfil de eventos (uma abordagem antiga); - Embora afirme que sua proposta seja de uso geral, ainda não conseguiu criar um descritor que fosse capaz de reconhecer todas as ações ou um conjunto de ações elementares.

Contexto de Aplicação do Trabalho		Autor	Breve descrição	Pontos Fortes	Críticas
Frameworks Computacionais Gerais	Sistema Perceptual e Motor	Siskind (2001)	Apresenta o sistema chamado LEONARD, que produz descrições fundamentais de eventos a partir de sequência de imagens.	-Diferente de modelos mais tradicionais, utiliza a relação da dinâmica de forças entre objetos para descrever eventos. Técnica mais robusta do que aquela que usa perfil do movimento; - Capaz de detectar sub-eventos (eventos hierárquicos); - Também fornece uma representação léxico-semântica para ação, fundamentada no ciclo perceptivo de evento.	- O modelo não se mostrou genérico o suficiente para explorar outras informações do evento que não fossem a ação e os objetos participantes.

Contexto de Aplicação do Trabalho		Autor	Breve descrição	Pontos Fortes	Críticas
Frameworks Computacionais Gerais	Planejamento	Narayanan (1999)	Trata-se de um modelo computacional para interpretar sentenças causais de domínio político e econômico e gerar inferências.	- Através de um tipo de redes de Petri modificado, é capaz de modelar informações aspectuais e temporais entre domínios encenados na sentença. Atua como se fosse um mecanismo de simulação mental robusto. (Isso explica porque tal sistema foi adaptado à GCC).	- Não é bidirecional, i.e., trata apenas a questão da interpretação de símbolos composicionais. - Embora use a simulação mental para gerar inferências situadas e incorporadas, não se sabe ao certo de onde o conhecimento usado para configurar as projeções metafóricas foi obtido. Isso nos leva a crer que o sistema ainda depende de seu projetista para funcionar .

Contexto de Aplicação do Trabalho		Autor	Breve descrição	Pontos Fortes	Críticas
Frameworks Computacionais Gerais	Planejamento	Roy (2005)	Framework teórico para guiar um mecanismo de planejamento a partir de interpretações de palavras e atos de fala.	- Trata o problema da fundamentação do símbolo utilizando simulações mentais, criadas a partir de rede de esquemas e signos; -Símbolos são manipulados sob a perspectiva da semiótica de Peirce;	- É uma arquitetura unidirecional, preocupando-se apenas com a interpretação dos símbolos composicionais; - Não trata a questão da aprendizagem de conceitos usados para fundamentar os esquemas e signos; - Além do mais, trata-se de apenas uma proposta que ainda não foi implementada.

Contexto de Aplicação do Trabalho		Autor	Breve descrição	Pontos Fortes	Críticas
Frameworks Computacionais Gerais	Planejamento	Frank e Vigliocco (2011)	Modelo conexionista para compreensão de sentenças baseado em simulação mental.	- Trata-se de uma proposta de formalização da teoria do sistema simbólico perceptual de Barsalou (1999). - Diferente de outras abordagens ele cria representações analógicas não somente para gerar inferências, mas também resolver ambiguidade.	- O sistema é unidirecional, i.e., só atua na interpretação dos símbolos composicionais; - As situações(conceitos) usadas para criar a simulação mental são representadas manualmente, o que gera certo grau de dependência do modelo com o projetista do sistema.

Contexto de Aplicação do Trabalho		Autor	Breve descrição	Pontos Fortes	Críticas
Frameworks Computacionais Aplicados à Linguística	Composicionalidade orientada à sintaxe	Vogt (2005)	Modelo para estudar a evolução e indução de estruturas composicionais na linguagem em uma população de agentes robóticos simulados, atuando no contexto dos jogos de linguagem.	- Consegue evoluir uma gramática comum aos agentes; - A gramática empregada favorece o surgimento de estruturas composicionais, inclusive estruturas holísticas. - Propõe diversos mecanismos para indução gramatical, tanto do ponto de vista de quem interpreta quanto de quem gera sentenças;	- Não se sabe se as técnicas de indução propostas seriam úteis para testar símbolos que envolvessem diferentes classes gramaticais, principalmente verbo, advérbios. Preocupa-se, principalmente, em aprender a estrutura da composição.

Contexto de Aplicação do Trabalho		Autor	Breve descrição	Pontos Fortes	Críticas
Frameworks Computacionais Aplicados à Linguística	Composicionalidade orientada à sintaxe	Sierra-Santibáñez (2014)	também utilizam jogos de linguagem para simular a evolução de um sistema linguístico para descrição de construções lógicas.	- O experimento aborda a aquisição gramatical, a partir da negociação em uma população de agentes (os agentes precisam entrar em acordo com relação à ordem adotada para formar uma sentença). Essa característica não é vista em trabalhos canônicos da literatura, uma vez que eles realizam o experimento entre 2 agentes apenas.	- Atua na perspectiva puramente simbólica da IA clássica, portanto acabam trazendo consigo o problema da fundamentação do símbolo. Os agentes conceituam as sentenças construindo fórmulas lógicas .

Contexto de Aplicação do Trabalho			Autor	Breve descrição	Pontos Fortes	Críticas
Frameworks Computacionais Aplicados à Linguística	Composicionalidade orientada à semântica	Modelos Inspirados na Neurociência	Glenberg e Gallese (2012)	Adaptaram dois modelos de controle motor (MOSAIC e HMOSAIC) para propor o modelo ABL (<i>action-based language</i>), capaz de aprender (substantivos, verbos e sentenças composicionais) compreender e produzir (gestos) linguagem.	- Parte do pressuposto de que o significado da linguagem surge como resultado de nossas interações situadas com o mundo , porém toma como base o sistema motor, onde a maioria dos trabalhos explora apenas o sistema perceptual; - Exploram o conceito de neurônios espelhos que vem da neurociência; - Trabalha tanto símbolos composicionais quanto isolados;	- Vocabulário limitado pela capacidade do sistema motor.

Contexto de Aplicação do Trabalho			Autor	Breve descrição	Pontos Fortes	Críticas
Frameworks Computacionais Aplicados à Linguística	Composicionalidade orientada à semântica	Gramática de Construções	Dominey <i>et al.</i> (2006)	Modelo para processamento de construções gramaticais.	- Modelo neuro-computacional que tenta descrever como humanos fazem o mapeamento entre forma e significado ao tentar interpretar uma sentença linguística;	- O modelo sígnico é diádico. Portanto, não há relação do símbolo com as coisas que ele designa no mundo. Acaba trazendo o problema da fundamentação do símbolo.
			Madden <i>et al.</i> (2010)	Modelo de Compreensão Híbrida que fundamenta construções e discursos em seus desdobramentos temporais provenientes de simulações situadas, controladas pela gramática.	- Trata o problema da fundamentação do símbolo apresentado pelo modelo neurolinguístico proposto por Dominey <i>et al.</i> (2006), ao propor o processamento de sentenças incorporadas; - Permite tratar cooperação e negociação entre robôs e humanos.	- Modelo é unidirecional, preocupa-se apenas com a interpretação dos símbolos composicionais. - Mais uma vez, os conceitos usados para gerar as simulações mentais não são aprendidos, dependendo de uma pré-configuração feita por seu projetista.

Contexto de Aplicação do Trabalho			Autor	Breve descrição	Pontos Fortes	Críticas
Frameworks Computacionais Aplicados à Linguística	Composicionalidade orientada à semântica	Gramática de Construções	Lallee <i>et al.</i> (2010)	Propôs uma extensão do trabalho de Madden <i>et al.</i> (2010), incluindo no framework a representação teleológica (raciocínio orientado a objetivos).	- A representação teleológica é uma componente crucial para o modelo, pois permite representar subcomponentes das ações, que inclui relações entre o estado inicial e final de sua execução, permitindo dessa forma, a interpretação de construções gramaticais mais complexas (conectivos causais, tipo porque, se- então).	- Mesma limitação destacada anteriormente.

Contexto de Aplicação do Trabalho			Autor	Breve descrição	Pontos Fortes	Críticas
Frameworks Computacionais Aplicados à Linguística	Composicionalidade orientada à semântica	Gramática de Construções Incorporadas	Bergen e Chang (2013)	Gramática de Construções Corporificadas (ECG - do inglês <i>Embodied Construction Grammar</i>) é um formalismo representacional de gramática que emprega modelos incorporados para sua aprendizagem e uso.	- Cobre várias particularidades de uma língua natural; - Os modelos incorporados provêm a semântica das construções através de inferências geradas dinamicamente.	- As simulações são pré-configuradas para poderem encenar eventos. A especificação semântica informa os parâmetros da simulação. Portanto, a simulação tem forte dependência com a gramática e com o corpus de construção gerado, assim como na FGC. - Não é bidirecional, trata apenas a interpretação simbólica.

Contexto de Aplicação do Trabalho			Autor	Breve descrição	Pontos Fortes	Críticas
Frameworks Computacionais Aplicados à Linguística	Composicionalidade orientada à semântica	Gramática de Construções Fluídas	Steels (2017)	A Gramática de Construções Fluídas (FCG - do inglês Fluid Construction Grammar) trata de uma plataforma computacional para desenvolver gramáticas de construções utilizadas em experimentos computacionais para estudar entendimento, produção e emergência de linguagem.	- As regras para gerar construções são consideradas bidirecionais e, portanto, utilizáveis tanto para análise quanto para produção simbólica.	- Trata-se de um feramental muito poderoso, porém complexo demais para aplicações onde não se deseja explorar todas as características da linguagem natural. - Depende de um sistema a parte para gerar as conceitualizações (sistema IRL).
			Spranger e Steels (2012)	Aplicação computacional usada para explicar a origem das estruturas composicionais, funções gramaticais e classes de palavras que descrevem relações espaciais.	- A gramática auxilia na escolha das estratégias de conceitualização que o ouvinte deve usar;	- As sentenças empregam apenas itens que descrevem localização espacial. Não é possível, por exemplo, verbo ou advérbios nas sentenças.

Contexto de Aplicação do Trabalho			Autor	Breve descrição	Pontos Fortes	Críticas
Frameworks Computacionais Aplicados à Linguística	Composicionalidade orientada à semântica	Gramática de Construções Fluídas	Steels et al. (2012)	Aplicação de jogos de linguagem, mais especificamente jogos de ação, para que agentes robóticos possam formular e interpretar comandos que especificam ações que o corpo pode realizar (por exemplo, levantar) ou ações que envolvam outros objetos de uma cena (por exemplo, pegar bloco vermelho).	- Adaptou ao sistema IRL redes semióticas para conceitualizar ações. - As ações abordadas são de diversas naturezas (ações monovalentes, bivalentes e trivalentes);	- Apresentaram resultados já alcançados por outros trabalhos. - Complexidade inerente aos recursos usados para processamento da semântica e sintaxe composicional (o sistema IRL e a gramática de construções fluidas). - Utilizam marcadores de caso (ou seja, sufixos anexados a palavras) em vez da ordem da palavras como os meios sintáticos para desambiguação semântica.

Contexto de Aplicação do Trabalho			Autor	Breve descrição	Pontos Fortes	Críticas
Frameworks Computacionais Aplicados à Linguística	Composicionalidade orientada à semântica	Gramática de Construções Fluídas	Trijp (2012)	Experimento para demonstrar a origem de uma gramática de casos. O trabalho demonstra experimentalmente a efetividade da gramática de casos para descrever eventos.	- Demonstração experimental das vantagens cognitivas de um sistema de casos.	- Totalmente dependente da gramática de construção fluida.

B. ASPECTOS DE IMPLEMENTAÇÃO DOS PACOTES DE SOFTWARE

Neste anexo são apresentados os aspectos de implementação dos 5 pacotes de software que compõem o *framework* computacional implementado.

Para facilitar a legibilidade do que apresentamos aqui, convencionou-se escrever os nomes de classes de software em negrito, iniciando sempre com letras maiúsculas. Já os métodos serão sempre descritos em negrito, porém iniciados com letras minúsculas, seguidos de “()” ao final. Os pacotes de software, quando citados, serão referenciados e os nomes descritos em itálico.

B.1 Aspectos de Implementação do Pacote *Raw*

O pacote *Raw*, como dito anteriormente, contém todas as classes responsáveis por captar as informações que são recebidas pelos sensores e formatá-las na mente do agente, de modo a formar uma representação bruta do objeto ou evento que será categorizado. Como mostra a Figura 7, ele é composto de três classes principais: **RawValue**, **RawValueGroup** e **RawObject**.

A classe **RawValue** é uma classe abstrata utilizada para definir um valor bruto para uma dimensão de qualidade. Por exemplo, o número real “25.0” pode representar o valor bruto atribuído para a dimensão de qualidade “graus celsius” que compõe o domínio da temperatura. Ela tem como atributos dois objetos do tipo **Double** que representam, respectivamente, o valor medido e um ruído que pode ter sido introduzido pela medição do sensor, além de um objeto do tipo **QualityDimension** ao qual um valor bruto estará associado. Complementando esses atributos, a classe possui o método abstrato **compare()** que pode ser configurado para comparar os valores brutos entre si.

A classe **RawValueGroup** é composta por um atributo *domínio* da classe **Domain** e uma lista de **RawValues**. Ela é utilizada para agrupar valores brutos de um domínio. Por exemplo, “255.0, 0.0, 0.0” é um valor bruto do domínio da cor, cujas dimensões de qualidade expressam “vermelho”, “verde” e “azul”.

A classe **RawObject**, por sua vez, representa a descrição bruta de um objeto por meio de seus **RawValueGroups**. Por este motivo, tem como atributo um **HashMap** que correlaciona a identificação do grupo de valores brutos e seus valores propriamente. Essas informações brutas são obtidas diretamente a partir dos sensores. Por exemplo, “cor: [255 (R), 0 (G), 0 (B)]; tamanho: [5 (X), 5 (Y), 5 (Z)]; forma: [6 (vértices)]” pode

ser o resultado dos sensores ao se perceber um cubo vermelho. Além desse atributo, a classe também possui o método **differenceTo()**, que calcula a diferença entre dois objetos brutos. Quanto maior o resultado da operação, maior será a diferença entre eles. Por fim, a classe também possui o método **fromProperties()**, que toma como base os protótipos das propriedades para montar um objeto bruto.

Além dos métodos mencionados, cada uma dessas classes é composta por seus devidos métodos construtores, **getters()** e **setters()**, que têm como principal objetivo instanciar seus objetos e facilitar o manuseio dos atributos e estruturas de dados, mantendo a integridade dos objetos instanciados. Dentre eles, um método que merece destaque é o **getDistance()** da classe **RawValueGroup**. Como a classe **RawValueGroup** é usada para definir grupos de valores brutos que formam um domínio, ela deve conter um método que servirá para medir a distância entre dois **RawValueGroups**. Sendo assim, este cálculo é feito pelo método **getDistance()**. Normalmente, a distância entre dois pontos p_1 e p_2 é feita através da distância euclidiana. No caso da implementação feita, utilizou-se a distância euclidiana ponderada, descrita pela equação B.1.

$$dist_E(p_1, p_2) = \sqrt{\sum_{i=1}^n w_i (p_{1,i} - p_{2,i})^2} \quad (B.1)$$

A opção por tal forma de cálculo permitiu conferir mais variabilidade à escolha das propriedades determinadas pelo espaço conceitual, levando em consideração, inclusive, aquelas formadas pela intersecção de duas regiões semânticas. Nesse caso, isso dependerá da ponderação atribuída a cada **RawValue**. Para entender como este método funciona, tomemos como exemplo um domínio de cor, determinado pelas dimensões de qualidade “tonalidade”, “saturação” e “brilho” (do inglês *HSB - Hue, Saturation and Brightness*) e as propriedades “vermelho”, “amarelo”, “escuro” e “claro” determinadas por pontos prototípicos e vetores de peso específicos. A propriedade vermelho, por exemplo, possui um ponto prototípico “0.0; 1.0; 1.0” e um vetor de pesos “1.0; 0.5; 0.5”. Já a propriedade escuro possui um ponto prototípico “0.0; 0.0; 0.2” e um vetor de pesos “0.0; 0.0; 1.0”. Os vetores de peso indicam que para a propriedade escuro, a única dimensão de qualidade importante é o brilho. Já para a propriedade vermelho todas as dimensões são importantes, porém a dimensão tonalidade é a mais importante. Assim, o método **detectProperties()** da classe **Domain**, ao receber o grupo de valores brutos “0.1; 1.0; 0.8”, calcula, utilizando a função **getDistance()**, a distância do mesmo em relação aos valores prototípicos de cada propriedade com seu respectivo vetor de pesos, e associa ao objeto todas as propriedades cuja distância estiver abaixo de um limite - neste caso, vermelho e escuro.

B.2 Aspectos de Implementação do Pacote *Conceptual*

O espaço conceitual, como visto no capítulo 2, é um *framework* semântico que permite a aprendizagem de conceitos. Ele tem como unidade elementar as dimensões de qualidade, que juntas (de maneira integral) formam um domínio. A distância calculada entre as entidades representadas no espaço conceitual permite medir o grau de similaridade entre elas. Tal métrica divide o domínio em regiões semânticas, que podem ser empregadas no processo de categorização. Sendo assim, um espaço conceitual pode ser visto como uma coleção de dimensões de qualidades divididas em domínios.

Na implementação proposta, o pacote *Conceptual* é composto pelas classes **Domain** e **QualityDimension**, mantendo referência direta com os dois principais elementos formadores do espaço conceitual, na definição de Gärdenfors.

A classe **QualityDimension** tem por objetivo representar uma dimensão de valores sensoriados, correspondendo a uma dimensão de qualidade. Ela tem como atributos dois valores **Double** que representam um intervalo de valores que uma dimensão pode assumir e um atributo *domínio*, da classe **Domain**, ao qual a dimensão de qualidade estará atrelada. Além desses atributos, a classe é composta por seu método construtor e métodos **getters** () e **setters**(), que assim como nas outras classes, irão facilitar o manuseio de seus atributos e estruturas de dados.

Já a classe **Domain** é formada por uma lista de dimensões de qualidade, que é utilizada para compor um domínio. Por exemplo, as dimensões “R, G e B” podem ser as dimensões de qualidade integrais que formam o domínio “cor”. Já o domínio “tamanho”, por sua vez, pode ser formado pelas dimensões de qualidade “altura, largura e profundidade”. Além desse atributo, ela também possui uma lista de propriedades e modificadores que correspondem às regiões semânticas identificadas durante o processo de segmentação do domínio e que foram mapeadas nessas categorias¹. Para tanto, a classe **Domain**, possui os métodos **detectProperties()** e **detectModifiers()**, que dado um **RawValueGroup** informado, irá verificar qual propriedade (ou modificador) pode representá-lo durante a categorização. Além desses dois métodos, há também os métodos construtores usados na instanciação de seus objetos e os métodos **getters()** e **setters()** usados na manipulação dos seus atributos.

As classes desse pacote têm uma função primordial durante a segmentação do espaço conceitual e mapeamento de suas regiões em categorias ontológicas. Para isso, elas, em conjunto, implementam um protocolo de aprendizagem que é responsável por descobrir os domínios e suas regiões semânticas a partir das informações vindas de diversos **RawObject** e seus **RawValueGroups**.

¹ Como será visto mais adiante no texto, a propriedade e o modificador são dois tipos de categorias ontológicas previstas no modelo.

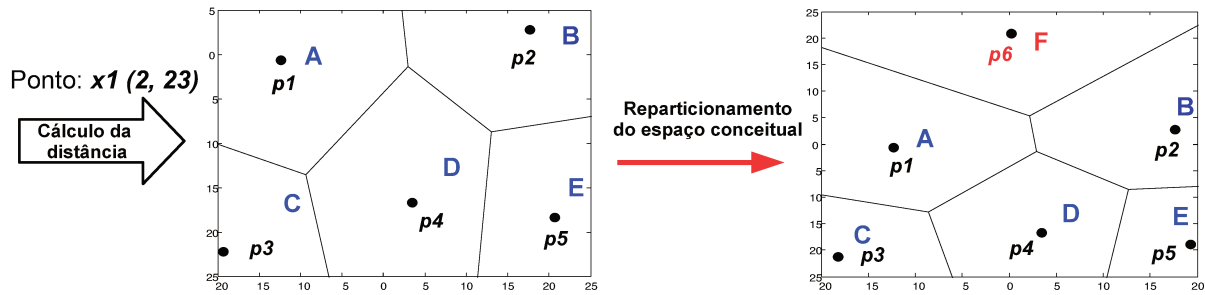


Figura 48 – Reparticionamento do espaço conceitual.

Sendo assim, uma vez que um **RawObject** foi sensoriado, cada um de seus **RawValueGroup** formadores são instanciados em objetos da classe **Domain**. Definir qual domínio instanciar fica a cargo dos próprios sensores que, neste caso, organizam as informações brutas em vetores, cujo tamanho está correlacionado às dimensões de qualidade correspondentes de cada domínio.

Como apresentado na seção anterior, o *framework* contempla uma forma específica para escolha das propriedades determinadas pelo espaço conceitual. Porém, nada impede que ele seja alterado, de modo que ele passe a selecionar as regiões semânticas determinadas por um algoritmo ou técnica de aprendizado de máquina. Neste caso, o algoritmo seria responsável por particionar o espaço conceitual e os métodos de detecção ficariam responsáveis por receber os valores brutos e correlacioná-los com a região semântica correspondente. Um exemplo deste mecanismo seria o que se segue: tomemos como base um domínio com duas dimensões de qualidade representado pelo diagrama de Voronoi à esquerda da Figura 48 e um ponto $x_1 (2, 23)$, representando um **RawValueGroup** qualquer. Cada região convexa do diagrama de Voronoi (regiões A, B, C, D e E) é representada por um ponto prototípico (p_1 , p_2 , p_3 , p_4 e p_5) que possui um raio de representatividade daquela região. Ao medir a distância entre cada ponto prototípico e o ponto $x_1 (2, 23)$, verifica-se que eles estão muito distantes no espaço. Sendo assim, nenhum ponto prototípico é capaz de representá-lo. Neste caso, se faz necessário reparticionar o domínio novamente, criando uma nova região para representá-lo (a região F), como mostra o diagrama mais à direita da Figura 48.

Novamente, tomando um novo ponto $x_2 (-8, 17)$, representando outro **RawValueGroup**, como mostra a Figura 49. Ao medir a distância entre ele e cada ponto prototípico das regiões atuais (regiões A, B, C, D, E e F), verifica-se que ele está próximo o suficiente da região E. Neste caso, o domínio não precisou ser segmentado. Mas o que aconteceu foi apenas um reposicionamento do ponto prototípico da região E, de modo que ele tenha melhor representatividade de todos os pontos que são convexas a essa região, inclusive o ponto x_2 .

Uma vez que as regiões semânticas do domínio são reconhecidas, elas podem ser mapeadas em propriedades. Portanto, toda vez que uma região semântica for encontrada,

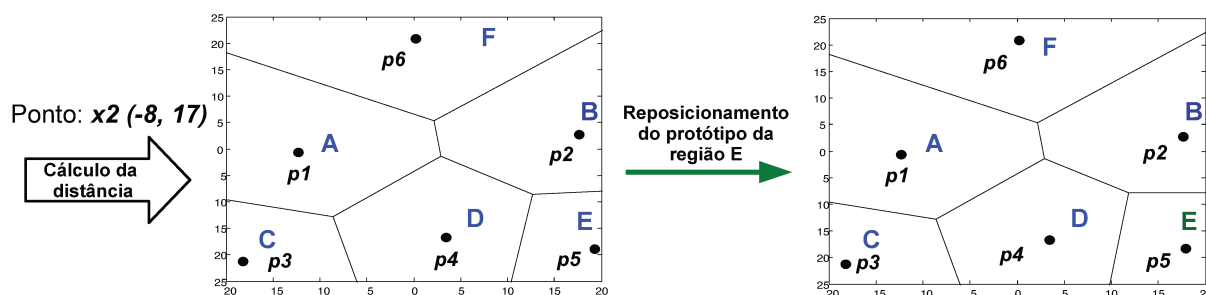


Figura 49 – Reposicionamento do ponto prototípico no espaço conceitual.

o algoritmo irá instanciar uma propriedade para referenciar semanticamente cada uma de suas regiões. Supondo que o diagrama de Voronoi das figuras anteriores representasse o domínio da posição, o ponto prototípico “F” representaria a posição “superior”, o ponto “A” representaria a posição “esquerda”, o ponto “B” representaria a posição “direita”, o ponto “D” representaria a posição “inferior”, o ponto “C” representaria a posição “inferior à esquerda” e o ponto “E” representaria a posição “inferior à direita”. É por este motivo que a classe **Domain** tem como um de seus atributos uma lista de propriedades.

Segundo o referencial teórico utilizado sobre Espaços Conceituais, se o mapeamento envolver apenas um domínio, a categoria gerada é uma propriedade. Mas se envolver mais de um domínio, trata-se de uma categoria de objeto. Sendo assim, para identificar as categorias de objetos, os métodos de detecção, junto com a técnica de aprendizado de máquina, poderiam, por exemplo, verificar a alta frequência com a qual um conjunto de propriedades esteja ocorrendo ao mesmo tempo. Todas as regiões ressaltadas e, representantes de uma propriedade, seriam demarcadas para juntas representar uma categoria de objeto.

Além de propriedades e categorias de objeto, o *framework* também deverá criar as demais categorias previstas no modelo. Neste caso, para a categoria *evento*, um método de detecção específico para ela poderia monitorar a memória de longo prazo, analisando as representações brutas dos objetos ao longo dos diferentes instantes de tempo, para verificar se uma de suas possibilidades de ação (neste trabalho nomeadas de *affordances*) foi aplicada. Para reconhecer uma *affordance*, uma técnica de aprendizado de máquina pode ser empregada para identificar a variação de mudanças de uma propriedade em um domínio específico e, em seguida, verificar se ela está ocorrendo mais de uma vez. Esse comportamento permite o aprendizado de uma nova ação. Para o caso da categoria *modificador*, o mecanismo atua de maneira diferente. Neste caso, ele gera um vetor resultado, fruto da comparação do valor bruto de uma propriedade antes e depois que uma *affordance* foi aplicada. No caso da categoria *episódio*, o *framework* mapeia e registra todo evento que envolve um objeto previamente selecionado. Maiores detalhes sobre cada uma das categorias ontológicas citadas serão apresentadas na subseção que se segue.

Vale ressaltar que o algoritmo responsável por realizar o mapeamento das

regiões semânticas de um ou mais domínios em categorias ontológicas é um tanto quanto sofisticado, pois para cada categoria ele precisa manipular o espaço conceitual de maneira distinta. Embora ele não esteja implementado em sua totalidade, testes iniciais já foram feitos e os resultados foram apresentados no capítulo 7.

B.3 Aspectos de Implementação do Pacote *Ontology*

Inspirado no artigo “A Teoria Semântica das Classes de Palavras” (GÄRDENFORS, 2014b) e na definição de ontologia da área de ciência da computação (GUARINO *et al.*, 2009), o pacote *Ontology* tem como objetivo representar um conjunto de categorias ontológicas, por meio do qual seja possível construir um modelo semântico de um fragmento espaço-temporal da realidade. A ideia é que a partir de um conjunto de instâncias dessas categorias ontológicas, seja possível elaborar uma simulação mental, no sentido de Barsalou (2009) e com isso criar um modelo semântico que possa ser correlacionado com sentenças de uma linguagem. Particularmente, as seguintes classes foram definidas: **Property**, **ObjectCategory**, **ConceptualObject**, **Affordance**, **Modifier**, **Event** e **Episode**. Cada classe permite que categorias ontológicas possam ser aprendidas a partir de regiões semânticas de um espaço conceitual, que servirão para fundamentar o significado de diferentes classes gramaticais usadas na linguagem.

A classe **Property** utiliza as regiões convexas de um domínio para representar as propriedades que integrarão o modelo semântico de adjetivos. Para tanto, uma propriedade sempre terá um atributo *domínio* que representa a região do espaço conceitual ao qual ela está associada. Esta classe também tem como um de seus atributos um **RawValueGroup** que é um ponto prototípico que melhor a representa. Vale ressaltar também que na nossa implementação pode haver intersecção entre uma ou mais regiões convexas de um domínio. Sendo assim, cada propriedade também possui como atributo uma lista de pesos que servem para determinar o quanto cada dimensão de qualidade será relevante na determinação da propriedade. Dessa forma, por exemplo, a propriedade *vermelho* pode se referir a qualquer cor (domínio formado pelas dimensões “R, G, B”) cuja dimensão de qualidade predominante seja a dimensão R. Além desses atributos, a classe **Property** tem métodos construtores para instanciação de seus objetos e métodos **getters()** e **setters()** para manipulação de seus atributos.

A classe **ObjectCategory** foi criada para definir o conceito de uma categoria de objetos e será nosso modelo semântico de substantivos. Seguindo Gärdenfors (2014b), categorias de objetos contêm informações de vários domínios e são usadas para que não precisemos nomear individualmente cada objeto. Sendo assim, neste trabalho uma categoria de objetos é formada por um atributo do tipo **HashMap** que representa um conjunto de domínios e suas respectivas propriedades, além de outros dois atributos do

tipo **ObjectCategory** que seriam suas partes composicionais e agregadoras, e três listas de **ObjectCategories** usadas para criar uma estrutura hierárquica entre categorias filhas, pais e outras categorias de objetos que podem conter objetos desta categoria, como mostra a Figura 50.

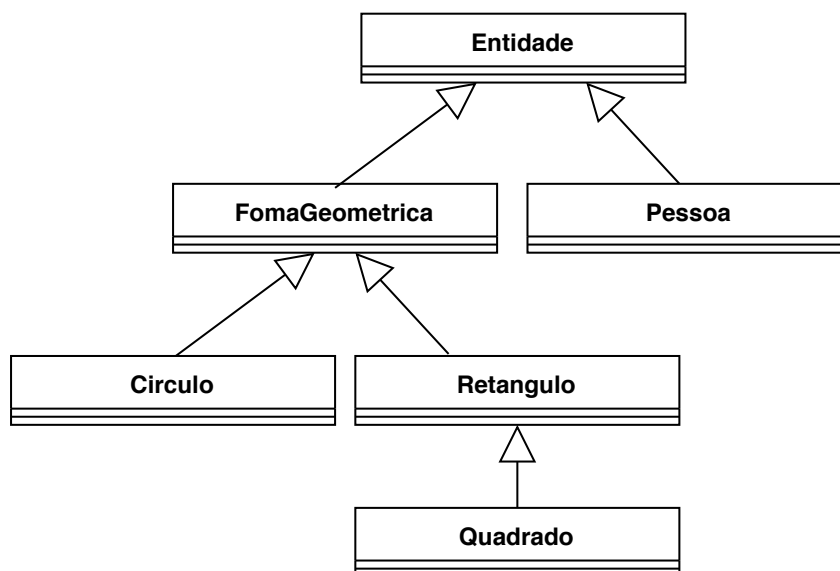


Figura 50 – Classificação dos conceitos de objetos modelados.

Um exemplo de uma categoria de objetos poderia ser uma classe **Carro**, cujos atributos seriam:

```

Propriedades: {
    Cores: [dourado, prata, branco, azul, preto, cinza, vermelho, laranja, verde, claro, escuro];
    Combustível: [Álcool, Gasolina, Diesel, Flex];
    Número de Rodas: [2, 4];
    Porta-Malas: [grande, médio e pequeno];
    Assentos: [3, 5, 15]
};
CategoriaPai: [Veículo];
CategoriaFilha: [Crossover, Esportivo, Hatch, Jipe, Picape, Sedan, Van];
PartesComposicionais: [chassi, carroceria, motor, rodas];
ContidoEm: [carreta, cegonha];
  
```

Neste caso, um Ford KA seria uma instância (ou exemplo) da categoria carro. Além desses atributos, a modelagem desta classe conta com métodos especiais, tais como o método **specialize()**, que dada uma categoria, permite adicioná-la como uma categoria filha em uma árvore hierárquica, o método **findCategory()**, que percorre a árvore hierárquica em busca de uma categoria específica e os métodos **addCompositionPart()** e **addAggregationPart()**, que adicionam partes composicionais e agregadoras nas categorias. Além desses métodos, existem também os métodos construtores para instanciação de seus objetos e métodos **getters()** e **setters()** para manipulação de seus atributos.

A classe **ConceptualObject** tem uma função similar à da classe **ObjectCategory**. Porém, ao invés de fundamentar os substantivos comuns, ela representará nosso modelo semântico de substantivos próprios. Segundo [Gärdenfors \(2014b\)](#), este conceito deve existir para atribuir um nome a uma entidade que mereça destaque. Ambas as categorias devem existir, pois nem todo conceito de objeto receberá um nome específico por motivos de economia de memória. Sendo assim, um objeto conceitual, além dos atributos e métodos citados na classe **ConceptualObject**, também terá como atributo um objeto **ObjectCategory** que representará a categoria à qual o objeto conceitual está associado. Um exemplo de objeto conceitual pode ser a entidade João, pertencente à categoria pessoa.

Para fundamentar a representação semântica dos verbos foi criada a classe **Affordance**. O termo “affordance” foi estabelecido por [Gibson \(1986\)](#) para designar possibilidades de ação disponíveis em objetos para que um indivíduo (independente de sua habilidade para perceber estas possibilidades) possa interagir com eles. Ao afirmar que todo objeto possui suas possibilidades de ação e que elas são independentes de quem interage com ele, [Gibson \(1986\)](#) corrobora com a teoria semântica das classes de palavras, pois para [Gärdenfors \(2014b\)](#) os verbos existem para descrever eventos que ocorrem sobre um objeto toda vez que uma de suas possibilidades de ação é aplicada. Neste trabalho, uma *affordance* irá modelar a semântica de verbos, por meio do registro do histórico de mudanças (ou variação) de uma propriedade de um domínio específico. Por exemplo, o verbo mover pode se referir a mudanças do domínio da posição. Já o verbo esquentar refere-se a mudanças no domínio da temperatura.

Na modelagem proposta, a classe **Affordance** é uma classe abstrata que possui como principal atributo o domínio ao qual ela está atrelada, ou seja, o domínio onde ocorre a mudança associada à *affordance*. Além disso, existem: duas categorias de objetos, que representam, respectivamente, o agente que pode aplicar esta *affordance* e o paciente que pode sofrê-la, a lista dos modificadores que ela admite (no parágrafo seguinte explicaremos com mais detalhes o que é e para que serve um modificador) e um **HashMap**, que associa, a cada modificador, o tempo que ele leva para se desenrolar. Além desses atributos, construtores e métodos **getters()** e **setters()** específicos, esta classe possui dois métodos abstratos que merecem destaque: o método **apply()**, que tem como função aplicar uma *affordance* toda vez que for necessário, e o método **detect()**, que tem como função verificar o histórico de mudanças de um domínio e detectar se uma *affordance* foi executada. Enquanto o método **detect()** é utilizado durante a percepção, o método **apply()** será usado durante a interpretação, para gerar uma reconstrução mental de uma cena vivenciada no passado. Como as mudanças conectadas com a *affordance* dependem especificamente de cada *affordance*, é necessário que cada sub-classe que estende a classe **Affordance** implemente, à sua maneira, esses dois métodos. Dessa forma, a opção por configurar a classe **Affordance** como abstrata, bem como também seus principais

métodos, se deu para flexibilizar a funcionalidade de cada *affordance* de acordo com o contexto ao qual ela está sendo aplicada. Ademais, como existe uma variedade de verbos possíveis em um idioma, seria impossível criar classes genéricas que pudessem atender a todos eles ao mesmo tempo. Uma outra abordagem possível seria implementar algoritmos de aprendizado de máquina que pudessem abstrair e aprender o comportamento dos verbos em questão. Entretanto, a complexidade de tal mecanismo extrapola o escopo do presente trabalho, sendo referenciada aqui como uma possibilidade para trabalhos futuros. Portanto, a classe **Affordance** deve ser configurada conforme sua aplicação antes da utilização do *framework* proposto.

A classe **Modifier** foi criada para definir o conceito do elemento que modifica o sujeito, objeto, ou verbo de uma sentença. Sendo assim, ela integra ao nosso modelo semântico o conceito de advérbios que modificam os verbos resultantes. No processo de categorização, esta classe será usada para comparar o valor bruto das propriedades do objeto que está sofrendo a ação, no início e no final de um evento, gerando um vetor resultante. Dessa maneira, ela possui um atributo **Domain** que representa o domínio ao qual o modificador está associado e onde ocorrerá uma mudança, um atributo **RawValueGroup** que é o vetor prototípico que melhor representa a região convexa do domínio que representa a mudança e uma lista de tipos que indica as operações que podem ocorrer no vetor resultante (e.g., a mudança pode ter sido simplesmente a geração de um valor resultante, ou uma multiplicação, ou uma soma, além de outras). Além desses atributos, cada dimensão do vetor resultante pode ser ponderada por uma lista de pesos, que irá determinar qual dimensão do vetor terá mais influência na determinação da mudança ocorrida. A classe **Modifier** tem ainda métodos construtores para a instanciação de seus objetos e métodos, além de **getters()** e **setters()** para manipulação de seus atributos.

Vale ressaltar que os advérbios que modificam os verbos de modo são modelados como uma especialização da classe **Affordance** correspondente. Sendo assim, a classe **Affordance**, além dos atributos citados, também possui dois atributos que representam o pai e filhos de uma *affordance*, sendo possível gerar também uma estrutura hierárquica entre elas. Assim, poderíamos ter, por exemplo, a *affordance* caminhar e, a partir de sua especialização, criar modificadores que alterariam o modo de caminhar, como por exemplo, caminhar em zig-zag, caminhar em curvas e outras maneiras.

A classe **Event** foi criada para representar nosso modelo semântico de estruturas mais complexas, tais como uma sentença, e será instanciada toda vez que uma possibilidade de ação/*affordance* for aplicada. Nesse modelo, uma sentença envolve um ou mais objetos e um único verbo, que modifica temporalmente alguma propriedade dos objetos envolvidos. Seguindo [Gärdenfors \(2014b\)](#), um evento pode ser formado a partir de um agente, uma ação, um paciente e um resultado, tal como João moveu o bloco amarelo rapidamente para a esquerda. Os agentes e pacientes são vistos como objetos com dife-

rentes propriedades. Sendo assim, devemos interpretá-los como instâncias de categorias de objetos (ou objetos conceituais, dependendo do contexto). Já a ação é a atuação do agente, exercendo uma força (ou uma sequência de vetores força), o que nós interpretamos como sendo uma *affordance*. E, por último, o resultado de um evento seria o vetor de mudança, representando a alteração das propriedades do agente e/ou paciente antes e depois da aplicação da *affordance*. Neste caso, interpretamos o resultado como um advérbio, representado semanticamente pela classe **Modifier**. Desta maneira, a classe **Event** tem como principais atributos dois conceitos de objetos que representam, respectivamente, o agente (João) e o paciente (bloco amarelo) do evento, uma *affordance* (mover), uma lista de modificadores que representam os modificadores presentes no evento (rapidamente e para a esquerda), dois **RawValueGroup**, representando os tempos inicial e final do evento e métodos construtores e métodos **getters()** e **setters()** para a manipulação desses atributos.

Por fim, a classe **Episode**, integra um conjunto de **Events** que possuem uma coesão temática comum, representando nosso modelo semântico para múltiplas sentenças concatenadas, que descrevem uma cena tematicamente coesa, como em um fragmento de texto mais longo, envolvendo mais de uma sentença. Uma instância de um **Episode** descreve um conjunto ordenado (ou paralelo, ou em superposição temporal parcial) de eventos centrados em um ou mais objetos que compõem o tema de uma cena. Um exemplo de episódio seria: João pegou a maçã. Maria cortava cebolas. João comeu a maçã. João cuspiu as sementes. Maria chorou. Para tanto, ela tem como atributos um conceito de objeto que representará o objeto foco e a lista de eventos relacionados ao objeto foco. Vale ressaltar que o objeto foco possuirá um raio de proximidade, e, neste caso, todos os eventos que ocorrerem com objetos que estiverem dentro deste raio estarão presentes no episódio em questão. Além dos atributos citados, esta classe também possuirá os métodos construtores e **getters()** e **setters()** usados na sua manipulação. Dessa forma, um episódio pode ser visto como uma simulação mental que representa semanticamente uma cena que pode ser descrita sintaticamente na forma de um conjunto ordenado de sentenças, que descreve o episódio.

B.4 Aspectos de Implementação do Pacote *Perception*

No pacote *perception* estão as classes responsáveis pelo sensoreamento, rastreamento e categorização dos objetos. Ele é composto pelas classes: **Mind**, **ImmediatePath**, **UnscentedKalmanFilter**, **CostThreshold**, **CostMatrix** e **HypergraphVertexCover**. Nos parágrafos que se seguem apresentaremos brevemente a função de cada classe. Na seção 5.2.1, onde apresentamos o algoritmo de rastreamento, está mais claro o contexto onde cada uma dessas classes pode ser aplicada, exceto a classe **Hypergraph-**

VertexCover, que é utilizada durante um jogo de linguagem, durante a discriminação dos objetos.

A classe **Mind** é um modelo da mente de um agente. Ela agrega todas as funcionalidades do *framework* e utiliza classes de todos os pacotes, inclusive as categorias ontológicas, para realizar o rastreamento de objetos, suas categorizações e o aprendizado linguístico.

A classe **ImmediatePath** é empregada para modelar caminhos (*paths*) de rastreamento dos objetos na memória imediata do agente. Cada caminho modelado por esta classe contém uma lista de passos (da subclasse *Step*).

A classe **UnscentedKalmanFilter** implementa a versão “*unscented*” do filtro de *Kalman* apresentada por [Julier e Uhlmann \(1997\)](#). Ela é empregada para remover ruídos dos valores brutos sensoreados e também para construir representações brutas estimadas para continuação de caminhos criados anteriormente na memória imediata.

A classe **CostThreshold** é uma classe auxiliar que foi criada apenas para armazenar os menores valores de um conjunto de números e indicar qual limiar separa esses valores dos demais. Os limiares calculados por esta classe são utilizados para eliminar caminhos ou passos que não estejam dentre os de menor custo.

Por fim, a classe **HypergraphVertexCover** implementa um algoritmo usado para encontrar a menor quantidade de “hiper-arestas” que cubram todos os vértices de um “hiper-grafo”. O agente a utiliza para escolher o menor conjunto de propriedades discriminativas para expressar simbolicamente um objeto conceitual.

B.5 Aspectos de Implementação do Pacote *Language*

A implementação proposta para o pacote de linguagem deve ser capaz de responder como os símbolos assumem ambas as funções gramaticais e sintáticas. Para isso, ela conta com regras genéricas que modelam uma gramática. Um modelo esquemático do diagrama de classes para este pacote pode ser observado na Figura 7 da página 83.

A classe **SymbolNode** define a construção de cada símbolo manipulado pelo mecanismo. Após a criação de um símbolo, ele fica armazenado nos nós de uma árvore, formando, assim, uma das principais estruturas de dados deste pacote de *software*. Cada nó da árvore pode ser visto como um símbolo completo ou não, uma vez que um símbolo pode ser dividido em partes. Quando ocorre a divisão de um símbolo em partes, essas ficam armazenadas entre nós pais e filhos. Essa estratégia foi adotada por questões de otimização da estrutura de dados que está se formando. Assim, por exemplo, se os símbolos “macaco” e “macarrão” forem armazenados na árvore, a parte inicial e comum entre eles “maca” ficará no nó pai, que terá dois nós filhos, um para armazenar a parte

“rrão” e o outro para armazenar a parte “co”. As folhas dessa árvore armazenam a categoria ontológica que faz a fundamentação do símbolo. Desta maneira, garantimos a associação entre um símbolo e seu significado. Mais adiante apresentaremos mais detalhes dessa estrutura de dados.

A classe **Media** descreve um tipo de mídia que pode ser utilizado em um sistema de comunicação. Neste caso, a mídia pode ser do tipo texto, áudio, sinal ou outras. Vale ressaltar que mídias distintas de um mesmo idioma serão tratados como sistemas de comunicação distintos. Esta classe contém métodos específicos que devem ser adaptados para manipular os símbolos da mídia que for adotada. Para o caso desse trabalho esses métodos foram configurados para manipular símbolos no formato texto.

A classe **Transformation** é usada para definir as transformações que podem ser aplicadas nos símbolos da linguagem que está em desenvolvimento. Entende-se como transformações as conjugações que um símbolo pode sofrer, tais como a transformação de tempo (presente, passado ou futuro), de número (singular ou plural) e de propriedade (aumentativo ou diminutivo). Atualmente, no modelo proposto, as transformações possíveis de serem empregadas são a de tempo (neste caso, passado e futuro) e a de número (somente a do tipo plural).

A classe **Language** é a classe mais importante do pacote, pois irá definir as configurações do mecanismo para que a proto-linguagem possa se desenvolver. Para tanto, ela agrega em sua estrutura um atributo de cada classe citada anteriormente, isto é, uma mídia, nós-símbolos e suas transformações, além das estruturas de dados que armazenam os pares (símbolo,categoria) e os métodos de evolução da linguagem, que permitirão com que símbolos possam ser criados e interpretados, bem como também as regras para reforçar o aprendizado gramatical e sintático.

Por fim, a classe **LanguageElement** faz a ligação entre os símbolos e as categorias ontológicas que eles designam. Dessa maneira, torna-se possível a associação entre um símbolo e seu significado, na forma de uma categoria ontológica².

² Do ponto de vista da semiótica Peirceana, o que ocorre aqui é a associação de um símbolo a um ícone, sendo que ambos têm em comum o mesmo objeto. Entendemos que nossa representação de uma categoria ontológica pode ser considerada uma metáfora, um tipo de ícone segundo a teoria Peirceana

C. DETALHES DOS EXPERIMENTOS REALIZADOS

C.1 Configuração das categorias ontológicas empregadas nos experimentos T1, T2 e T3

Domínio: TIME_DOMAIN

		Ponto Prototípico da Dimensão de Qualidade (Peso)
Classe Ontológica	Nome da instância	TIME_DIMENSION
Modificador	TIME_MODIFIER_SLOWLY	5.0 (1.0)
	TIME_MODIFIER_NEUTRAL	2.0 (1.0)
	TIME_MODIFIER_QUICKLY	0.0 (1.0)

Domínio: POSITION_DOMAIN

		Ponto Prototípico da Dimensão de Qualidade (Peso)	
Classe Ontológica	Nome da instância	POSITION_DIMENSION_X	POSITION_DIMENSION_Y
Propriedade	POSITION_PROPERTY_NORTH	0.0 (0.0)	5.0 (1.0)
	POSITION_PROPERTY_SOUTH	0.0 (0.0)	-5.0 (1.0)
	POSITION_PROPERTY_EAST	5.0 (1.0)	0.0 (0.0)
	POSITION_PROPERTY_WEST	-5.0 (1.0)	0.0 (0.0)
	POSITION_PROPERTY_NEUTRAL	0.0 (1.0)	0.0 (0.0)
Modificador	POSITION_MODIFIER_UP	0.0 (0.0)	2.0 (1.0)
	POSITION_MODIFIER_DOWN	0.0 (0.0)	-2.0 (1.0)
	POSITION_MODIFIER_LEFT	-2.0 (1.0)	0.0 (0.0)
	POSITION_MODIFIER_RIGHT	2.0 (1.0)	0.0 (0.0)
	POSITION_MODIFIER_NEUTRAL_X	0.0 (1.0)	0.0 (0.0)
	POSITION_MODIFIER_RIGHT_NEUTRAL_Y	0.0 (0.0)	0.0 (1.0)

Domínio: DIRECTION_DOMAIN

		Ponto Prototípico da Dimensão de Qualidade (Peso)
Classe Ontológica	Nome da instância	DIRECTION_DIMENSION
Propriedade	DIRECTION_PROPERTY_UP	90.0 (1.0)
	DIRECTION_PROPERTY_LEFT	180.0 (1.0)
	DIRECTION_PROPERTY_DOWN	270.0 (1.0)
	DIRECTION_PROPERTY_RIGHT	0.0 (1.0)
Modificador	DIRECTION_MODIFIER_CLOCKWISE	-20.0 (1.0)
	DIRECTION_MODIFIER_NEUTRAL	0.0 (1.0)
	DIRECTION_MODIFIER_COUNTERCLOCKWISE	20.0 (1.0)

Domínio: SIZE_DOMAIN

		Ponto Prototípico da Dimensão de Qualidade (Peso)
Classe Ontológica	Nome da instância	SIZE_DIMENSION
Propriedade	SIZE_PROPERTY_SMALL	1.0 (1.0)
	SIZE_PROPERTY_NEUTRAL	5.0 (1.0)
	SIZE_PROPERTY_BIG	10.0 (1.0)
Modificador	SIZE_MODIFIER_SMALLER	-1.0 (1.0)
	SIZE_MODIFIER_SAME	0.0 (1.0)
	SIZE_MODIFIER_BIGGER	1.0 (1.0)

Domínio: COLOR_DOMAIN		Ponto Prototípico da Dimensão de Qualidade (Peso)		
Classe Ontológica	Nome da instância	COLOR_ DIMENSION_ HUE	COLOR_ DIMENSION_ SATURATION	COLOR_ DIMENSION_ BRIGHTNESS
Propriedade	COLOR_PROPERTY_RED	0.0 (1.0)	1.0 (0.5)	1.0 (0.5)
	COLOR_PROPERTY_YELLOW	1/6 (1.0)	1.0 (0.5)	1.0 (0.5)
	COLOR_PROPERTY_GREEN	1/3 (1.0)	1.0 (0.5)	1.0 (0.5)
	COLOR_PROPERTY_CYAN	1/2 (1.0)	1.0 (0.5)	1.0 (0.5)
	COLOR_PROPERTY_BLUE	2/3 (1.0)	1.0 (0.5)	1.0 (0.5)
	COLOR_PROPERTY_PURPLE	5/6 (1.0)	1.0 (0.5)	1.0 (0.5)
	COLOR_PROPERTY_BLACK	0.0 (0.0)	0.0 (0.0)	0.0 (1.0)
	COLOR_PROPERTY_WHITE	1.0 (0.0)	0.0 (1.0)	1.0 (1.0)
	COLOR_PROPERTY_BRIGHT	0.0 (0.0)	0.0 (0.0)	1.0 (0.8)
	COLOR_PROPERTY_DARK	0.0 (0.0)	0.0 (0.0)	0.2 (1.0)
	COLOR_PROPERTY_ NEUTRAL_BRIGHTNESS	0.0 (0.0)	0.0 (0.0)	0.5 (1.0)
	COLOR_PROPERTY_SATURATED	0.0 (0.0)	1.0 (1.0)	1.0 (0.5)
	COLOR_PROPERTY_ NEUTRAL_SATURATION	0.0 (0.0)	0.5 (1.0)	0.0 (0.0)
	COLOR_PROPERTY_GREY	0.0 (0.0)	0.0 (1.0)	0.5 (0.5)
Modificador	COLOR_MODIFIER_RED	0.0 (1.0)	1.0 (0.1)	1.0 (0.1)
	COLOR_MODIFIER_YELLOW	1/6 (1.0)	1.0 (0.1)	1.0 (0.1)
	COLOR_MODIFIER_GREEN	1/3 (1.0)	1.0 (0.1)	1.0 (0.1)
	COLOR_MODIFIER_CYAN	1/2 (1.0)	1.0 (0.1)	1.0 (0.1)
	COLOR_MODIFIER_BLUE	2/3 (1.0)	1.0 (0.1)	1.0 (0.1)

Domínio: COLOR_DOMAIN

		Ponto Prototípico da Dimensão de Qualidade (Peso)		
Classe Ontológica	Nome da instância	COLOR_ DIMENSION_ HUE	COLOR_ DIMENSION_ SATURATION	COLOR_ DIMENSION_ BRIGHTNESS
	COLOR_MODIFIER_PURPLE	5/6 (1.0)	1.0 (0.1)	1.0 (0.1)
Modificador	COLOR_MODIFIER_DARKER	0.0 (0.0)	0.0 (0.0)	-0.3 (1.0)
	COLOR_MODIFIER_ NEUTRAL_BRIGHTNESS	0.0 (0.0)	0.0 (0.0)	0.0 (1.0)
	COLOR_MODIFIER_ MORE_SATURATED	0.0 (0.0)	0.3 (1.0)	0.0 (0.0)
	COLOR_MODIFIER_ NEUTRAL_SATURATION	0.0 (0.0)	0.0 (1.0)	0.0 (0.0)
	COLOR_MODIFIER_GRAYER	0.0 (0.0)	-0.3 (1.0)	0.0 (0.0)
	COLOR_MODIFIER_BRIGHTER	0.0 (0.0)	0.0 (0.0)	0.3 (1.0)

Domínio: SHAPE_DOMAIN

		Ponto Prototípico da Dimensão de Qualidade (Peso)				
Classe Ontológica	Nome da instância	SHAPE_ DIMENSION_ _VERTEX _NUMBER	SHAPE_ DIMENSION_ _EDGE _NUMBER	SHAPE_ DIMENSION_ _WH _RATIO	SHAPE_ DIMENSION_ _MIN _ANGLE	SHAPE_ DIMENSION_ _MAX _ANGLE
Propriedade	SHAPE_ PROPERTY_ EDGELESS	0.0 (1.0)	0.0 (1.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)

Domínio: SHAPE_DOMAIN		Ponto Prototípico da Dimensão de Qualidade (Peso)				
Classe Ontológica	Nome da instância	SHAPE_ DIMENSION _VERTEX _NUMBER	SHAPE_ DIMENSION _EDGE _NUMBER	SHAPE_ DIMENSION _WH _RATIO	SHAPE_ DIMENSION _MIN _ANGLE	SHAPE_ DIMENSION _MAX _ANGLE
	SHAPE_ PROPERTY_ TRILATERAL	3.0 (1.0)	3.0 (1.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
Propriedade	SHAPE_ PROPERTY_ QUADRI LATERAL	4.0 (1.0)	4.0 (1.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
	SHAPE_ PROPERTY_ PERSON	0.0 (0.0)	1000.0 (1.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
	SHAPE_ PROPERTY_ RING	0.0 (0.0)	2000.0 (1.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
	SHAPE_ PROPERTY_ WIDE	0.0 (0.0)	0.0 (0.0)	2.0 (1.0)	0.0 (0.0)	0.0 (0.0)
	SHAPE_ PROPERTY_ TALL	0.0 (0.0)	0.0 (0.0)	0.5 (1.0)	0.0 (0.0)	0.0 (0.0)

Domínio: SHAPE_DOMAIN

		Ponto Prototípico da Dimensão de Qualidade (Peso)				
Classe Ontológica	Nome da instância	SHAPE_ DIMENSION _VERTEX _NUMBER	SHAPE_ DIMENSION _EDGE _NUMBER	SHAPE_ DIMENSION _WH _RATIO	SHAPE_ DIMENSION _MIN _ANGLE	SHAPE_ DIMENSION _MAX _ANGLE
	SHAPE_ PROPERTY_ SKEWED	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	60.0 (1.0)	120.0 (1.0)
Propriedade	SHAPE_ PROPERTY_ RIGHT_ ANGLED	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	90.0 (1.0)	90.0 (1.0)
	SHAPE_ PROPERTY_ NEUTRAL_ RATIO	0.0 (0.0)	0.0 (0.0)	1.0 (1.0)	0.0 (0.0)	0.0 (0.0)
	SHAPE_ PROPERTY_ NEUTRAL	0.0 (1.0)	0.0 (0.0)	0.0 (0.0)	0.0 (1.0)	0.0 (1.0)

Affordances
AFFORDANCE_MOVE
AFFORDANCE_PAINT

Categoria de Objeto	Especializa Categoria Pai	Propriedade
CATEGORY_ PERSON		SHAPE_PROPERTY_PERSON
		SHAPE_PROPERTY_NEUTRAL_RATIO
		SHAPE_PROPERTY_NEUTRAL
CATEGORY_ PERSON		POSITION_DIMENSION_X
		POSITION_DIMENSION_Y
		COLOR_PROPERTY_RED
		COLOR_PROPERTY_BLUE
		COLOR_PROPERTY_NEUTRAL_SATURATION
		COLOR_PROPERTY_NEUTRAL_BRIGHTNESS
		DIRECTION_DIMENSION
CATEGORY_ GEOMETRIC_ SHAPE		SHAPE_PROPERTY_EDGELESS
		SHAPE_PROPERTY_TRILATERAL
		SHAPE_PROPERTY_QUADRILATERAL
		SHAPE_PROPERTY_WIDE
		SHAPE_PROPERTY_TALL
		SHAPE_PROPERTY_SKEWED
		SHAPE_PROPERTY_RIGHT_ANGLED
		SHAPE_PROPERTY_NEUTRAL_RATIO
		SHAPE_PROPERTY_NEUTRAL
		POSITION_DIMENSION_X
		POSITION_DIMENSION_Y
		SIZE_PROPERTY_SMALL
		SIZE_PROPERTY_NEUTRAL
		SIZE_PROPERTY_BIG

Categoria de Objeto	Especializa Categoria Pai	Propriedade
		COLOR_PROPERTY_RED
		COLOR_PROPERTY_YELLOW
		COLOR_PROPERTY_GREEN
		COLOR_PROPERTY_CYAN
CATEGORY_ GEOMETRIC_ SHAPE		COLOR_PROPERTY_BLUE
		COLOR_PROPERTY_PURPLE
		COLOR_PROPERTY_BLACK
		COLOR_PROPERTY_WHITE
		COLOR_PROPERTY_BRIGHT
		COLOR_PROPERTY_DARK
		COLOR_PROPERTY_NEUTRAL_BRIGHTNESS
		COLOR_PROPERTY_SATURATED
		COLOR_PROPERTY_NEUTRAL_SATURATION;
		COLOR_PROPERTY_GREY
CATEGORY_ RING		SHAPE_PROPERTY_RING
		SHAPE_PROPERTY_NEUTRAL_RATIO
		SHAPE_PROPERTY_NEUTRAL
		POSITION_DIMENSION_X
		POSITION_DIMENSION_Y
		SIZE_PROPERTY_SMALL
		SIZE_PROPERTY_NEUTRAL
		SIZE_PROPERTY_BIG
		COLOR_PROPERTY_RED
		COLOR_PROPERTY_YELLOW

Categoria de Objeto	Especializa Categoria Pai	Propriedade
		COLOR_PROPERTY_GREEN
		COLOR_PROPERTY_CYAN
		COLOR_PROPERTY_BLUE
		COLOR_PROPERTY_PURPLE
		COLOR_PROPERTY_BLACK
CATEGORY_ RING		COLOR_PROPERTY_WHITE
		COLOR_PROPERTY_BRIGHT
		COLOR_PROPERTY_DARK
		COLOR_PROPERTY_NEUTRAL_BRIGHTNESS
		COLOR_PROPERTY_SATURATED
		COLOR_PROPERTY_NEUTRAL_SATURATION
		COLOR_PROPERTY_GREY
CATEGORY_ CIRCLE	CATEGORY_ GEOMETRIC_ SHAPE	SHAPE_PROPERTY_EDGELESS
		SHAPE_PROPERTY_SKEWED
		SHAPE_PROPERTY_NEUTRAL_RATIO
		POSITION_DIMENSION_X
		POSITION_DIMENSION_Y
		SIZE_PROPERTY_SMALL
		SIZE_PROPERTY_NEUTRAL
		SIZE_PROPERTY_BIG
		COLOR_PROPERTY_RED
		COLOR_PROPERTY_YELLOW
		COLOR_PROPERTY_GREEN
		COLOR_PROPERTY_CYAN

Categoria de Objeto	Especializa Categoria Pai	Propriedade
		COLOR_PROPERTY_BLUE
		COLOR_PROPERTY_PURPLE
		COLOR_PROPERTY_BLACK
		COLOR_PROPERTY_WHITE
		COLOR_PROPERTY_BRIGHT
		COLOR_PROPERTY_DARK
CATEGORY_ CIRCLE	CATEGORY_ GEOMETRIC_ SHAPE	COLOR_PROPERTY_NEUTRAL_BRIGHTNESS
		COLOR_PROPERTY_SATURATED
		COLOR_PROPERTY_NEUTRAL_SATURATION;
		COLOR_PROPERTY_GREY
CATEGORY_ TRIANGLE	CATEGORY_ GEOMETRIC_ SHAPE	SHAPE_PROPERTY_TRILATERAL
		SHAPE_PROPERTY_WIDE
		SHAPE_PROPERTY_TALL
		SHAPE_PROPERTY_SKEWED
		SHAPE_PROPERTY_NEUTRAL_RATIO
		POSITION_DIMENSION_X
		POSITION_DIMENSION_Y
		SIZE_PROPERTY_SMALL
		SIZE_PROPERTY_NEUTRAL
		SIZE_PROPERTY_BIG
		COLOR_PROPERTY_RED
		COLOR_PROPERTY_YELLOW
		COLOR_PROPERTY_GREEN
		COLOR_PROPERTY_CYAN

Categoria de Objeto	Especializa Categoria Pai	Propriedade
		COLOR_PROPERTY_BLUE
		COLOR_PROPERTY_PURPLE
		COLOR_PROPERTY_BLACK
		COLOR_PROPERTY_WHITE
		COLOR_PROPERTY_BRIGHT
		COLOR_PROPERTY_DARK
		COLOR_PROPERTY_NEUTRAL_BRIGHTNESS
CATEGORY_ RECTANGLE	CATEGORY_ GEOMETRIC_ SHAPE	COLOR_PROPERTY_SATURATED
		COLOR_PROPERTY_NEUTRAL_SATURATION;
		COLOR_PROPERTY_GREY
		SHAPE_PROPERTY_QUADRILATERAL
		SHAPE_PROPERTY_WIDE
		SHAPE_PROPERTY_TALL
		SHAPE_PROPERTY_RIGHT_ANGLED
		SHAPE_PROPERTY_NEUTRAL_RATIO
		POSITION_DIMENSION_X
		POSITION_DIMENSION_Y
		SIZE_PROPERTY_SMALL
		SIZE_PROPERTY_NEUTRAL
		SIZE_PROPERTY_BIG
		COLOR_PROPERTY_RED
		COLOR_PROPERTY_YELLOW
		COLOR_PROPERTY_GREEN
		COLOR_PROPERTY_CYAN

Categoria de Objeto	Especializa Categoria Pai	Propriedade
		COLOR_PROPERTY_BLUE
		COLOR_PROPERTY_PURPLE
		COLOR_PROPERTY_BLACK
		COLOR_PROPERTY_WHITE
		COLOR_PROPERTY_BRIGHT
		COLOR_PROPERTY_DARK
		COLOR_PROPERTY_NEUTRAL_BRIGHTNESS
		COLOR_PROPERTY_SATURATED
CATEGORY_ RECTANGLE	CATEGORY_ GEOMETRIC_ SHAPE	COLOR_PROPERTY_NEUTRAL_SATURATION;
		COLOR_PROPERTY_GREY
CATEGORY_ SQUARE	CATEGORY_ RECTANGLE	SHAPE_PROPERTY_QUADRILATERAL
		SHAPE_PROPERTY_RIGHT_ANGLED
		SHAPE_PROPERTY_NEUTRAL_RATIO
		POSITION_DIMENSION_X
		POSITION_DIMENSION_Y
		SIZE_PROPERTY_SMALL
		SIZE_PROPERTY_NEUTRAL
		SIZE_PROPERTY_BIG
		COLOR_PROPERTY_RED
		COLOR_PROPERTY_YELLOW
		COLOR_PROPERTY_GREEN
		COLOR_PROPERTY_CYAN
		COLOR_PROPERTY_BLUE

Categoria de Objeto	Especializa Categoria Pai	Propriedade
		COLOR_PROPERTY_PURPLE
		COLOR_PROPERTY_BLACK
		COLOR_PROPERTY_WHITE
		COLOR_PROPERTY_BRIGHT
		COLOR_PROPERTY_DARK
		COLOR_PROPERTY_NEUTRAL_BRIGHTNESS
		COLOR_PROPERTY_SATURATED
		COLOR_PROPERTY_NEUTRAL_SATURATION;
		COLOR_PROPERTY_GREY
CATEGORY_ TRAPEZIUM	CATEGORY_ GEOMETRIC_ SHAPE	SHAPE_PROPERTY_QUADRILATERAL
		SHAPE_PROPERTY_SKEWED
		SHAPE_PROPERTY_WIDE
		SHAPE_PROPERTY_TALL
		SHAPE_PROPERTY_NEUTRAL_RATIO
		POSITION_DIMENSION_X
		POSITION_DIMENSION_Y
		SIZE_PROPERTY_SMALL
		SIZE_PROPERTY_NEUTRAL
		SIZE_PROPERTY_BIG
		COLOR_PROPERTY_RED
		COLOR_PROPERTY_YELLOW
		COLOR_PROPERTY_GREEN
		COLOR_PROPERTY_CYAN
		COLOR_PROPERTY_BLUE

Categoria de Objeto	Especializa Categoria Pai	Propriedade
		COLOR_PROPERTY_PURPLE
		COLOR_PROPERTY_BLACK
		COLOR_PROPERTY_WHITE
		COLOR_PROPERTY_BRIGHT
		COLOR_PROPERTY_DARK
		COLOR_PROPERTY_NEUTRAL_BRIGHTNESS
		COLOR_PROPERTY_SATURATED
		COLOR_PROPERTY_NEUTRAL_SATURATION;
		COLOR_PROPERTY_GREY

C.2 Interface Gráfica da Simulação

A interface gráfica implementada para a aplicação dos jogos de linguagem pode ser visualizada na Figura 51.

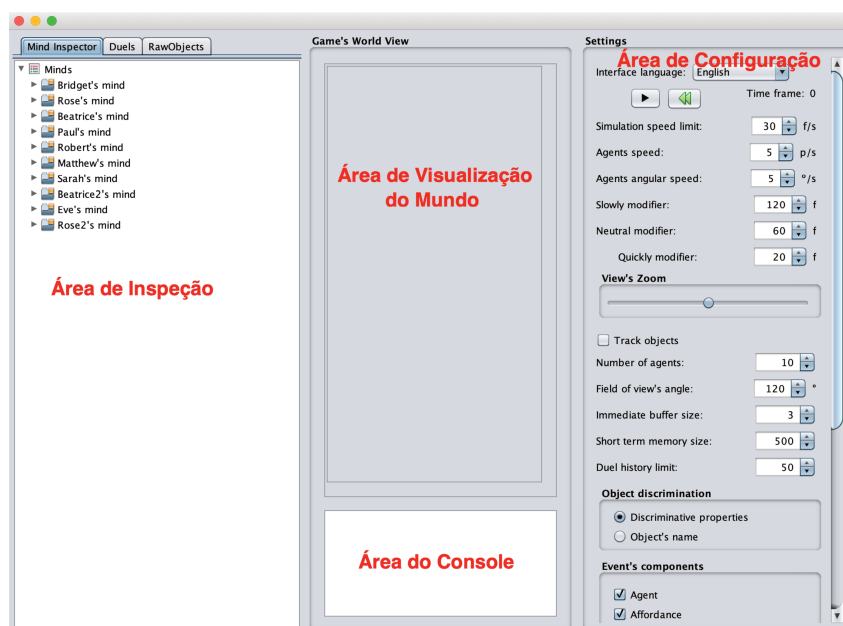


Figura 51 – Interface da Aplicação do Jogo de Linguagem.

Ela é composta por quatro áreas distintas:

- Uma área para visualização do mundo simulado;
- Uma área para informar a configuração do jogo;
- Uma área para inspeção da mente dos agentes, dos duelos e dos objetos brutos;
- Uma área para o console, onde os símbolos trocados durante os duelos são registrados.

Nos parágrafos que se seguem falamos sobre cada área separadamente.

C.2.1 A Área de Visualização do Mundo

A área de visualização do mundo é o local onde a execução do jogo pode ser visualizada. Nesta área os agentes estão dispostos em fila, aguardando serem selecionados para um duelo. Além dos agentes, ela também possui um ringue onde os objetos estão dispostos e sobre os quais os agente executam eventos. Próximo ao ringue existe uma área na qual os agentes (falante e ouvinte) deverão se posicionar. Durante um duelo, os agentes devem seguir o seguinte protocolo de movimentação dentro dessa área:

1. Os agentes escolhidos saem da fila de espera e se dirigem às posições de falante e ouvinte. O falante se posiciona na ponta à esquerda e abaixo do ringue e o ouvinte se posiciona à direita e abaixo do ringue. Ao se posicionarem, os dois agentes se viram para o ringue;
2. O falante emite os símbolos desejados. O ouvinte se dirige à posição de controle. A posição de controle fica abaixo do ringue e ao centro. Ao se posicionar, o ouvinte vira para o ringue e executa o evento que interpretou;
3. Após terminar a execução, o ouvinte retorna à sua posição. Caso o evento executado não seja o que o falante imaginou inicialmente, o falante se dirige à posição de controle e executa um evento de correção;
4. Após o ouvinte reforçar as associações dos símbolos corretos, os dois agentes retornam às suas posições na linha de espera.

C.2.2 A Área de Configuração

A área de configuração é onde devem ser informadas as configurações desejadas para a execução do jogo. Para uma visualização dessa área, veja a Figura 52.

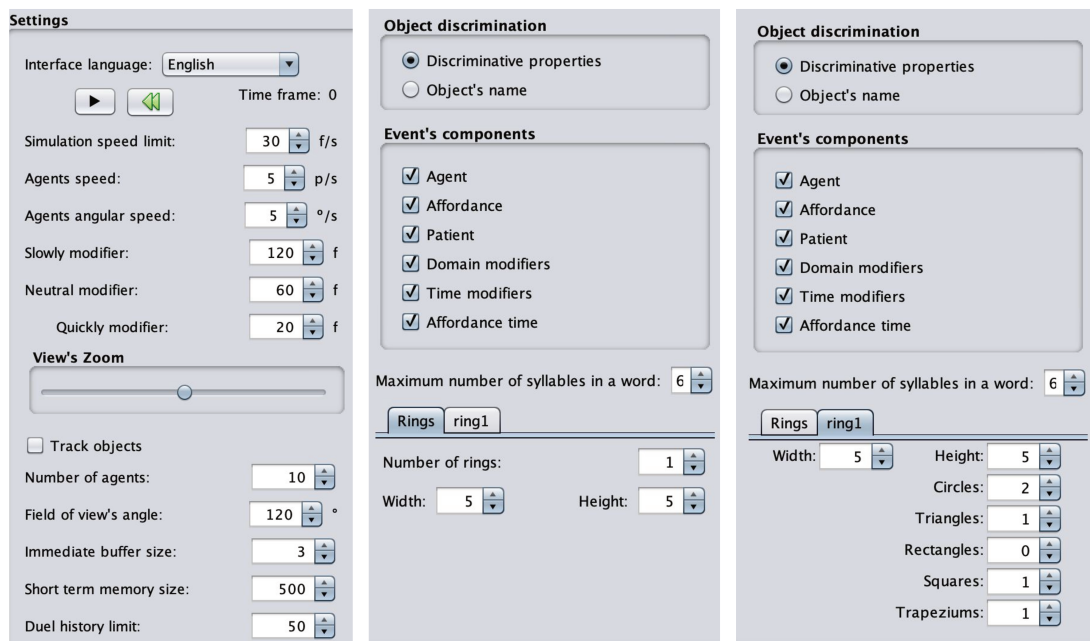


Figura 52 – Área de Configuração.

A primeira informação que pode ser configurada é a linguagem dos textos contidos na interface, onde é possível escolher entre os idiomas inglês (o padrão) ou português. Existe também um botão para iniciar/pausar a execução do jogo. A configuração de execução da simulação pode ser feita através das seguintes opções:

- *Simulation speed limit*: define o número máximo de *frames* que devem ser simulados a cada segundo, derivando, assim, a velocidade de execução dos jogos;
- *Agents speed*: define a velocidade em que o agente se movimenta. Essa medida é dada em *pixels* por segundo;
- *Agents angular speed*: define a velocidade em que o agente gira. Essa medida é dada em graus por segundo;
- *Slowly/Neutral/Quickly modifiers*: configuram o número de *frames* necessários para a execução dos modificadores no domínio de tempo. O maior valor configura o modificador *slowly* (lentamente), o menor valor configura o modificador *quickly* (rapidamente) e o valor médio entre esses dois configura o modificador *neutral* (tempo neutro/normal);
- *View's Zoom*: controle usado para aproximar (ou ampliar) o conteúdo exibido na área de visualização;
- *Number of agents*: configura o tamanho da população de agentes que irá participar do jogo;
- *Field of view's angle*: configura em graus o ângulo do campo de visão do agente;
- *Track objects*: essa configuração permite habilitar/desabilitar o algoritmo de rastreamento proposto. Como este algoritmo envolve várias etapas de processamento, ele pode trazer limitações de desempenho para o jogo. Assim, nos jogos onde o rastreamento não é importante, pode-se usar uma versão simplificada desse algoritmo;
- *Immediate buffer size*: configura o tamanho da memória imediata usada no algoritmo de rastreamento apresentado na seção [B.4](#);
- *Short term memory size*: configura o tamanho da memória de curto prazo usada no algoritmo de rastreamento apresentado na seção [B.4](#);
- *Duel History Limit*: configura o tamanho do histórico de duelos passados presente na área de inspeção, na aba de duelos. A quantidade informada neste campo também configurará a quantidade de duelos cujos símbolos estarão presentes na área do console.

A interface também prevê opções para configurar como os objetos serão discriminados e escolher os elementos que irão compor o evento. As seguintes combinações de componentes do evento são possíveis:

- *Agent* ou *Patient* discriminado por nome: é estabelecido um jogo de adivinhação. São empregados objetos conceituais para fundamentar o significado dos símbolos trocados no jogo;
- *Patient* discriminado por propriedades: ocorrerá um jogo de linguagem composicional. Os símbolos usados para formar a sentença designarão categorias de objetos e propriedades.
- *Affordance*: ocorrerá um jogo de ação. Uma *affordance* será usada para fundamentar o significado dos símbolos trocados no jogo;
- *Agent, Patient, Affordance, Domain modifiers, Time Modifiers, Affordance time*: ocorrerá um jogo de linguagem composicional. Assim, sentenças composicionais serão expressas, cujos símbolos remetem às categorias agente, paciente, *affordance* e modificadores. A sentença conterá também modificadores de tempo (lentamente, rapidamente e tempo neutro) e nas *affordances* poderão ser aplicadas as transformações de passado e futuro¹.

Por fim, ainda é possível informar na área de configuração o número máximo de sílabas que os símbolos podem ter, bem como a quantidade de ringues e seus tamanhos.

Para cada ringue é possível escolher a quantidade de objetos que estarão disponíveis para uso nos duelos. Existem 5 categorias de objetos distintas: Círculo, Triângulo, Retângulo, Quadrado e Trapézio.

C.2.3 A Área de Inspeção

A área de inspeção foi criada para consultar a mente dos agentes, os duelos estabelecidos e os objetos brutos durante a execução do jogo.

Na aba correspondente à mente, várias informações relevantes podem ser obtidas. Como mostra a Figura 53, para cada agente da população é possível verificar os objetos que ele categorizou, os eventos que ele percebeu e suas linguagens.

Ao expandir o nó correspondente aos objetos, como mostra o item A da Figura 54, é possível verificar as propriedades e partes que compõem os objetos categorizados.

Neste mesmo nó há também uma referência à memória de curto prazo, que contém uma lista com todas as informações brutas daquele objeto em cada *frame* em que ele foi sensoreado. Essas informações são relevantes para a categorização do objeto e identificação das *affordances* que ele sofreu e de seus modificadores.

¹ As transformações de futuro e passado previstas para este jogo de linguagem ocorreram da seguinte forma: na transformação de futuro será acrescida a terminação “f” ao final do símbolo que designa a *affordance*, enquanto na transformação de passado será acrescida a terminação “p”.

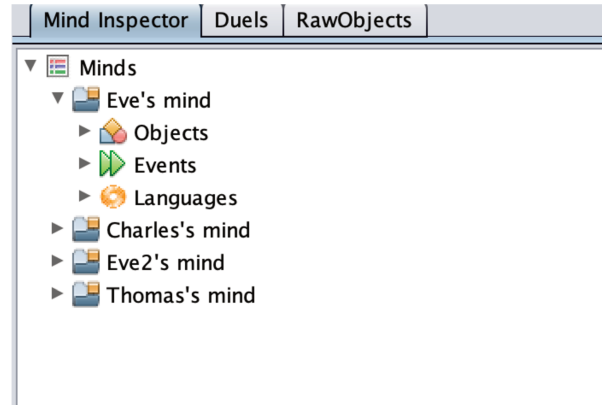


Figura 53 – Área de Inspeção da Mentis.

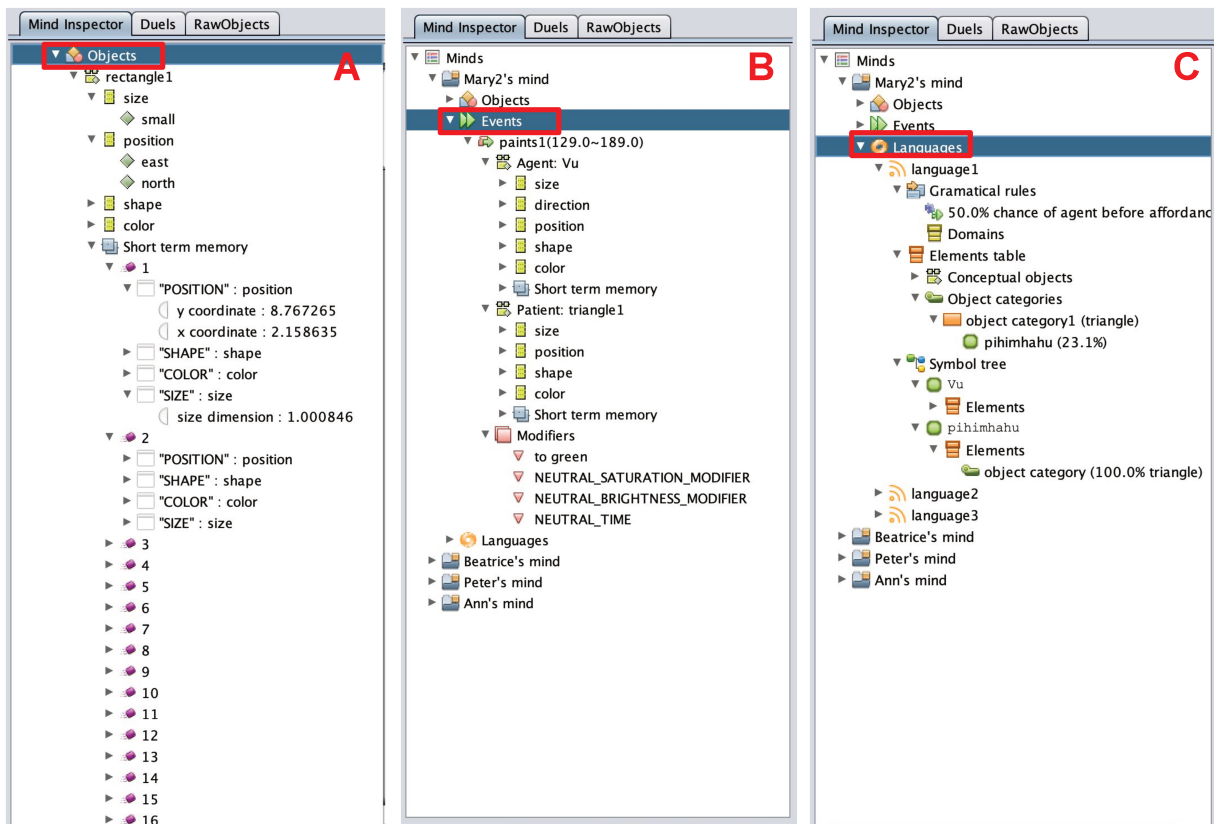


Figura 54 – Apresentação dos objetos categorizados (A), dos eventos percebidos (B) e da linguagem que está em evolução durante o jogo (C).

No item B da Figura 54, no nó correspondente aos eventos, é possível verificar uma lista com todas as *affordances* que foram aplicadas. Ao lado do nome do evento há a indicação da sequência de *frames* em que ele ocorreu. O total de *frames* indica a duração do evento, que está relacionado ao modificador de tempo. Para cada evento, além da *affordance*, há a indicação do agente e paciente envolvidos e também dos modificadores aplicados.

Já no nó correspondente à linguagem é possível verificar todas as linguagens do agente, como mostra o item C da Figura 54. No caso dos experimentos que estamos

validando, uma linguagem está sendo testada (a *language1*). É possível verificar a evolução de suas regras gramaticais e a composição da tabela de categorias e da árvore de símbolos.

A tabela de categorias irá correlacionar cada categoria prevista pelo modelo com os símbolos que estão disputando para representá-la e o percentual que cada símbolo possui na disputa. Já árvore correlaciona um símbolo com suas categorias, registrando a probabilidade de uma categoria fazer sua fundamentação. A representação gráfica de como essas estruturas são representadas na interface pode ser vista na Figura 55.

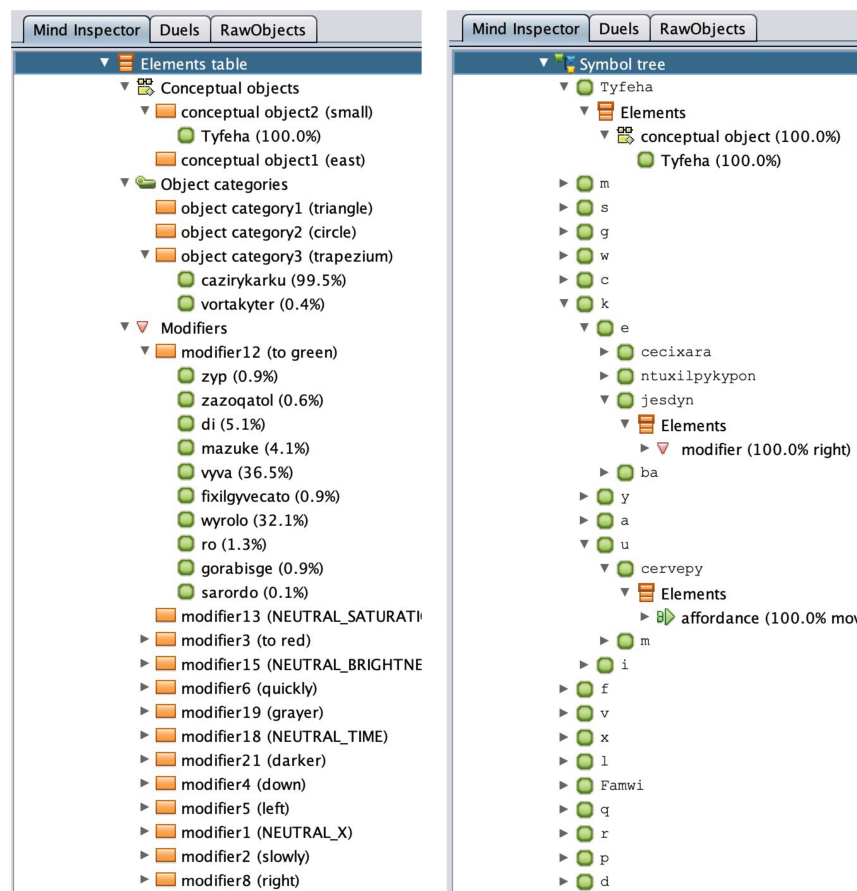


Figura 55 – Estruturas de dados expandidas. O item da esquerda exhibe a tabela de categorias e o item da direita a árvore de símbolos.

Repare que os nomes em inglês utilizados na interface são apenas para facilitar o entendimento da depuração. Eles não se referem aos símbolos empregados pelos agentes em sua linguagem.

Na área de inspeção também existe uma aba para conferir os duelos que ocorrem nos diferentes ringues do ambiente, como apresenta a instância A da Figura 56. Para cada ringue, há informações a respeito do duelo corrente e sobre duelos passados.

No caso do duelo corrente (instância B da Figura 56), há indicação dos agentes que desempenham o papel de falante e ouvinte, o status de suas movimentações no

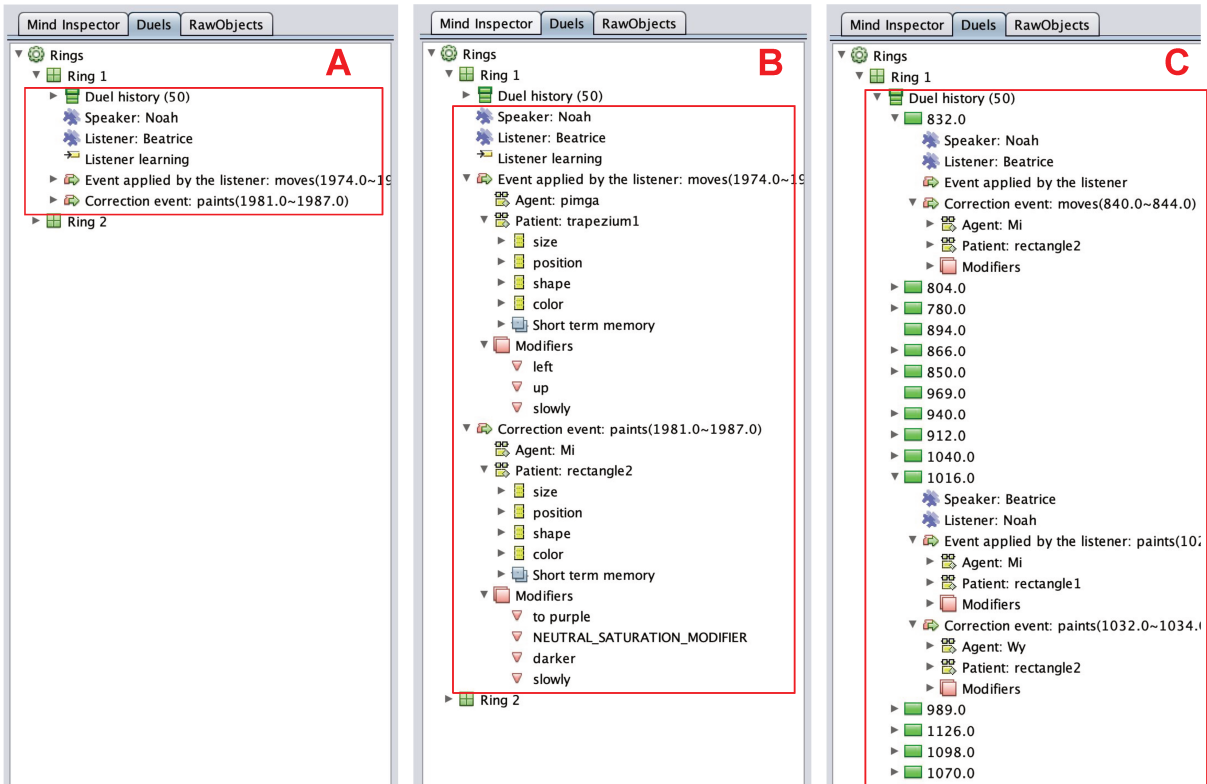


Figura 56 – Área de Inspeção. Em (A) aba de duelos e suas informações. Em (B) as informações do duelo corrente. E em (C) as informações do histórico de duelos.

ringue durante o transcorrer do duelo, o evento que foi aplicado pelo ouvinte na tentativa de interpretar o que foi dito pelo falante e, no caso de erro, o evento de correção executado pelo falante. Em ambos os eventos há indicação de *affordance*, agente, paciente e modificadores aplicados.

Já no caso dos duelos passados (instância C da Figura 56), esses podem ser conferidos no histórico de duelos. Essa área foi criada para facilitar a depuração do jogo. Os duelos lá armazenados são referenciados pelo instante de tempo em que iniciaram e contém todas as informações de como ocorreram (o agente, paciente, a *affordance* aplicada e seus modificadores, o evento aplicado pelo ouvinte e, se for o caso, o evento de correção). O histórico pode ser configurado para armazenar quantos duelos forem necessários.

Por fim, também existe na área de inspeção uma aba para conferir os dados dos objetos brutos, neste caso os agentes da população, ringues e as figuras geométricas que eles contém. Como mostra a Figura 57, essas informações são provenientes do processo de simulação e, por esse motivo, refletem o status corrente dos objetos. Essas informações servem para validar, por exemplo, se uma *affordance* foi aplicada corretamente.

Cada uma das abas presentes na área de inspeção tem como objetivo apresentar informações válidas para auxiliar a depuração do jogo que está se desenrolando.

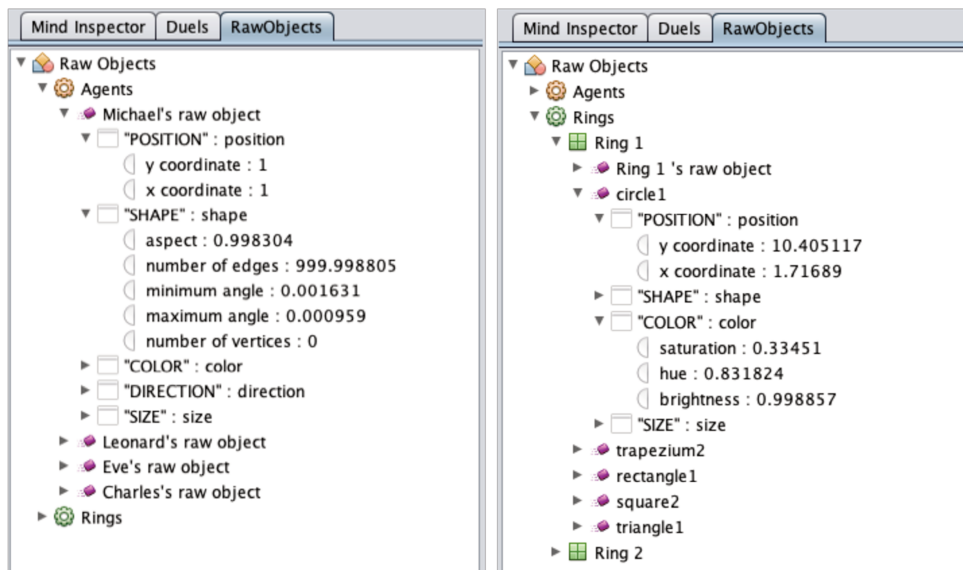


Figura 57 – Área de Inspeção: A aba de objetos brutos.

C.2.4 A Área de Console

A área de console foi criada com o objetivo de visualizar as sentenças emitidas pelo agente falante durante o duelo. Conforme as opções configuradas (como os componentes do evento), o agente falante recupera os símbolos correspondentes, cria uma sentença e a expressa. Após expressa, a sentença pode ser conferida na área de console, como exhibe a Figura 58.

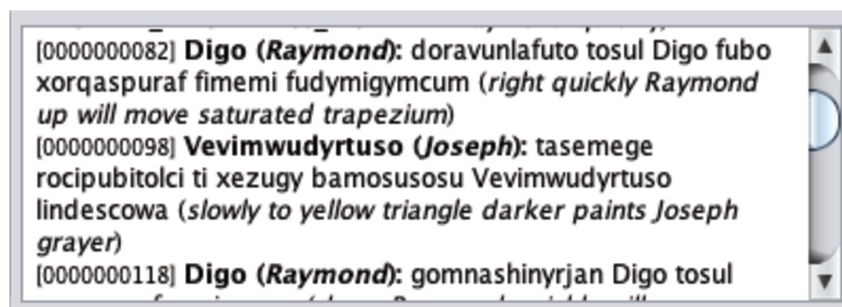


Figura 58 – A área de console com as sentenças expressas.

Como a língua dos agentes é própria e não tem relação com nenhuma língua natural, foi criado um mecanismo que a traduz para inglês ou português (conforme o idioma indicado na área de configuração), para facilitar o entendimento das sentenças trocadas durante o jogo. Vale ressaltar que não há influência da tradução nos símbolos e regras gramaticais que estão evoluindo durante o jogo.

Como pode ser observado na Figura 58, as informações da área de console são apresentadas no seguinte formato: entre “[]” o instante de tempo em que a sentença foi expressa, em negrito o nome do agente falante e após os “:” a sentença dita. As informações contidas entre “()” dizem respeito à tradução da sentença. Vale ressaltar também,

que o console possui uma capacidade semelhante à do histórico de duelos passados. Assim, se ele for configurado para apresentar 50 duelos, por exemplo, após atingir esse limite, essa área é atualizada com as novas sentenças ditas, removendo-se as mais antigas.