

*Modelagem Computacional Educacional Abdutiva:  
concepção e investigação experimental da atividade*

**Ricardo José Martins**

Junho / 2017

Dissertação de Mestrado em Ciência da  
Computação

# **Modelagem Computacional Educacional Abdutiva: concepção e investigação experimental da atividade**

Esse documento corresponde à Dissertação apresentada à Banca Examinadora para a defesa de Mestrado em Ciência da Computação da Faculdade Campo Limpo Paulista.

Membros da banca:

---

Prof. Dr. Osvaldo Luiz de Oliveira  
(Orientador, FACCAMP)

---

Profª. Dra. Ana Maria Monteiro  
(FACCAMP)

---

Prof. Dr. Jacques Duílio Brancher  
(DC-UEL)

---

Prof. Dr. Marcelo de Paiva Guimarães  
(Suplente, FACCAMP)

Campo Limpo Paulista, 23 de junho de 2017.

## FICHA CATALOGRÁFICA

Dados Internacionais de Catalogação na Publicação (CIP)

Câmara Brasileira do Livro, São Paulo, Brasil.

Martins, Ricardo José

Modelagem computacional educacional abduativa: concepção e investigação experimental da atividade / Ricardo José Martins. Campo Limpo Paulista, SP: FACCAMP, 2017.

Orientador: Prof. Dr. Osvaldo Luiz de Oliveira

Dissertação (Programa de Mestrado em Ciência da Computação) – Faculdade Campo Limpo Paulista – FACCAMP.

1. Modelagem computacional. 2. Construtivismo. 3. Abdução. 4. Raciocínio abduativo. 5. Linguagem *Constraint Handling Rules*. I. Oliveira, Osvaldo Luiz de. II. Campo Limpo Paulista. III. Título.

CDD-624.171

## AGRADECIMENTOS

A Deus, por me acompanhar sempre em todos os momentos da minha vida.

À minha mãe Neuza, e ao meu pai Pedro, que sempre me incentivam a buscar meus sonhos.

Ao meu orientador, o professor Dr. Osvaldo Luiz de Oliveira, pelos ensinamentos, pela paciência e por ter acreditado na potencialidade deste trabalho.

Ao Instituto Federal de Educação, Ciência e Tecnologia do Sul de Minas – IFSULDEMINAS – Campus Muzambinho, com o auxílio financeiro através do PIQ (Programa de Incentivo à Qualificação).

Aos meus colegas e ex-colegas de trabalho do IFSULDEMINAS.

Aos meus ex-professores da graduação, que me deram a base de conhecimento para seguir minha carreira na área de tecnologia da informação.

Aos meus amigos, que nunca deixaram de estar presentes em minha vida.

Aos meus colegas do curso de mestrado da FACCAMP, que nesses 3 anos dividiram comigo as dificuldades, mas também muitas alegrias e companheirismo.

Aos professores e funcionários do programa de Mestrado em Ciência da Computação da FACCAMP.

A todos os alunos que participaram da pesquisa, doando uma parte de seu tempo, e a todos os demais alunos que tive a honra de conviver até o momento, e aos muitos mais que ainda terei esta mesma honra, de hoje em diante.

**“O mais desperdiçado de todos os dias é aquele em que nós não rimos.”**

(Sébastien-Roch Nicolas de Chamfort)

**Resumo:** *Modelagem computacional refere-se à atividade de criar programas que estabelecem modelos para fenômenos, i.e., simulam em computador fenômenos reais ou abstratos. No campo da Educação, a modelagem tem sido utilizada como uma opção didática para implementar um paradigma educacional construtivista, no qual o estudante é convidado a aprender através de atividades de desenvolvimento de modelos computacionais para, por exemplo, máquinas, circuitos, fenômenos biológicos, físicos etc.. A ideia fundamental é a de ofertar ao estudante uma linguagem de programação e motivá-lo a se colocar em um ciclo no qual, por exemplo, ele desenvolve um programa que simula algo (e.g., uma máquina, um fenômeno), executa o programa, interpreta os resultados da execução e, se os resultados são inconsistentes com o funcionamento do que está sendo modelado, ele reformula o programa desenvolvido, na medida em que também reconstrói o seu conhecimento sobre o que está sendo modelado. Abdução é um tipo de raciocínio que objetiva formular hipóteses para explicar fatos observados, considerando uma teoria como fundamento. Todavia, raciocínios abdutivos não têm sido utilizados no contexto da modelagem computacional educacional, o que motiva a proposta de uma nova modalidade de modelagem, que se distingue das demais pelo foco na programação de raciocínios abdutivos. Especificamente, este trabalho propõe (1) diretrizes que fundamentam as ideias e os procedimentos sobre como raciocínios abdutivos podem ser empregados em modelagem computacional educacional e (2) a investigação experimental da atividade de modelagem de raciocínios abdutivos com propósitos educacionais. Os resultados sugerem a viabilidade das diretrizes propostas e indicam importantes desafios técnicos e pedagógicos a serem superados.*

**Palavras-chave:** Modelagem Computacional, Construtivismo, Abdução, Raciocínio Abdutivo, Linguagem *Constraint Handling Rules* (CHR).

***Abstract:** Computational modeling is the activity of creating programs that establish models for phenomena, i.e., simulate real or abstract phenomena in computer. In the education field, modeling has been used as a didactic option to implement a constructivist educational paradigm, in which the student is invited to learn via activities of development of computational models for, for example, machines, circuits, biological and physical phenomena, etc.. The fundamental idea is to offer the student a programming language and motivate him to engage in a cycle in which, for example, he develops a program that simulates something (e.g., a machine, a phenomenon), executes the program, interprets the results of the execution and, if the results are inconsistent with the what is being modeled, it reformulates the developed program, and also reconstructs its knowledge about what is being modeled. Abduction is a type of reasoning that aims to formulate hypotheses to explain observed facts, considering a theory as a foundation. However, abductive reasoning has not been used in the context of educational computational modeling, and this motivates the proposal of a new modality of modeling, which is distinguished from the others by the focus on the abductive reasoning programming. Specifically, this work proposes (1) a guideline that defines the ideas and procedures for educational computational modeling of abductive reasoning and (2)–an experimental investigation of the activity of modeling abductive reasoning for educational purposes. The results suggest the feasibility of the proposed guideline and indicate important technical and pedagogical challenges to be overcome.*

**Keywords:** Computational Modeling, Constructivism, Abduction, Abductive Reasoning, Constraint Handling Rules (CHR) Language.

## **Glossário**

ACLP	Abductive Constraint Logic Programming
ALP	Abductive Logic Programming
CHR	Constraint Handling Rules
CLP	Constraint Logic Programming
IQON	Interacting Quantities Omitting Numbers
LPO	Lógica de Primeira Ordem
MbL	Modeling-based Learning
STELLA	Structural Thinking Experimental Learning Laboratory with Animation

# Sumário

Capítulo 1 Introdução.....	1
1.1 – O que é modelagem computacional educacional? .....	2
1.2 – O que é abdução?.....	3
1.3 – Objetivos deste trabalho .....	4
1.4 – Organização deste documento .....	5
Capítulo 2 Modelagem Computacional Educacional .....	7
2.1 – Modelagem computacional educacional: origem e evolução .....	8
2.2 – Ambiente para modelagem computacional típico, atores e papéis desempenhados....	10
Capítulo 3 Abdução .....	13
3.1 – Uma definição formal de raciocínio abduutivo .....	14
3.2 – Exemplo de formalização de raciocínios abdutivos .....	15
3.3 – Critérios para escolha de boas hipóteses.....	17
Capítulo 4 Metodologia.....	19
4.1 – Visão geral da metodologia.....	20
4.2 – Descrição das etapas metodológicas .....	21
Capítulo 5 Proposta de diretrizes para Modelagem Computacional Educacional Abdutiva .....	23
5.1 – O que são as diretrizes no contexto deste trabalho?.....	24
5.2 – Diretrizes para modelagem computacional educacional abdutiva.....	24
5.3 – Exemplo de uso das diretrizes propostas.....	27
Capítulo 6 Linguagem CHR: introdução à programação e uso para abdução.....	30
6.1 – Introdução .....	31
6.2 – Regras CHR .....	32
6.3 – Semântica operacional de um programa CHR .....	33
6.4 – Uso da linguagem CHR para implementar raciocínio abduutivo .....	34
Capítulo 7 Pesquisa Experimental sobre Atividade de Modelagem Computacional Educacional Abdutiva .....	40
7.1 – Visão geral da metodologia da pesquisa.....	41
7.2 – Materiais.....	43
7.3 – Procedimento experimental.....	46
7.4 – Sujeitos .....	48
7.5 – Critérios para análise quantitativa e qualitativa dos resultados da pesquisa experimental .....	48
7.6 – Resultados e discussão .....	50
Capítulo 8 Discussão e Conclusão .....	54

8.1 – Contribuições deste trabalho .....	55
8.2 – Conclusões.....	56
8.3 – Consequências das conclusões e trabalhos futuros.....	57
Apêndice I Formulário: pesquisa experimental sobre modelagem computacional educacional abduativa.....	58
Apêndice II Padrão de Resposta aos Exercícios da Pesquisa Experimental .....	69
Apêndice III Artigos publicados e em processo de avaliação.....	80
Referências.....	81

# Capítulo 1

## Introdução

*Este trabalho é sobre modelagem computacional educacional abduativa. Assim, as duas primeiras seções deste capítulo objetivam explicar o que isto significa respondendo a duas questões: “O que é modelagem computacional educacional?” (Seção 1.1) e “O que é abdução?” (Seção 1.2). A Seção 1.3 discorre sobre os objetivos deste trabalho e, por fim, a Seção 1.4 descreve a organização, em capítulos, deste documento.*

## 1.1 – O que é modelagem computacional educacional?

Modelagem é o processo de construir modelos, que podem ser, de exemplo, objetos, paisagens, edifícios, circuitos, máquinas, processos, conceitos abstratos, fenômenos etc.. O processo de modelagem, que conduz ao estabelecimento de modelos, está presente quando:

- (1) Uma criança desenha à lápis, sobre um papel, uma casa, em um cenário contendo árvores e animais domésticos.
- (2) Um estudante de educação pré-escolar utiliza massa de modelagem para moldar a figura tridimensional de uma joaninha.
- (3) Um jovem monta um quebra-cabeça de 1000 peças para composição de uma foto, ou usa peças LEGO (Lego, 2017) para desenvolver o protótipo de um avião.
- (4) Um escultor usa uma pedra para esculpir a estátua de alguma celebridade.
- (5) Um aluno de eletrônica emprega uma placa Arduino (Arduino-Org, 2017), *leds*, resistores, cabos, e uma *protoboard* para desenvolver um circuito que imita o comportamento de um semáforo de trânsito.
- (6) Um escritor retrata, em português, todos os detalhes de uma paisagem campestre.
- (7) Um matemático propõe um conjunto de equações para resolver um problema de otimização combinatória.

Diferentes tipos de modelos ocorrem em cada uma das situações descritas. Por exemplo, o modelo em (1) são os traços à lápis sobre um papel, em (2) é a representação da joaninha via massa de modelar e em (5) é o circuito (*protoboard*, *leds*, resistores, placa Arduino e cabos), juntamente com o programa (*script*) que controla o funcionamento do circuito. Diferentes também são os artefatos e técnicas usados na modelagem. Por exemplo, em (1) os artefatos são o lápis e o papel, e a técnica de modelagem consiste em desenhar à mão livre traços sobre o papel de maneira que o desenho represente os principais componentes da casa e da paisagem. Em (4) os artefatos englobam a pedra, martelos e cinzéis, e a técnica de modelagem poderia ser a da cinzelação.

A modelagem computacional é um tipo de modelagem em que o modelo é estabelecido em um programa de computador. A possibilidade de executar o programa em um computador, dá “vida” ao modelo. Modelos computacionais variam desde simples

desenhos bidimensionais ou tridimensionais, até complexas simulações, por exemplo, jogos eletrônicos, previsão de tempo, síntese de proteínas e voo de aeronaves.

A modelagem computacional educacional é o tipo de modelagem computacional que se caracteriza pela criação de modelos em computador com o objetivo de um estudante aprender sobre o que é modelado. Não se trata de aprender sobre computadores, linguagens de programação e programação, mas aprender sobre o assunto, objeto, processo, conceito ou fenômeno a ser modelado pelo estudante. A ideia de desenvolver um ambiente educacional para aprendizagem via modelagem tem origem nos trabalhos de Papert na década de 1960 (Papert, 1993). Basicamente, um ambiente para modelagem computacional oferece uma linguagem de programação para o estudante descrever um modelo, e envolve um ciclo no qual o estudante desenvolve um programa que simula algo (uma máquina, um processo, um conceito etc.), executa o programa, interpreta os resultados da execução e, se os resultados da execução são inconsistentes com o que ele está modelando, o estudante altera o programa desenvolvido. Este processo no qual programas são desenvolvidos e alterados permite ao estudante aprender sobre o que ele está modelando.

## **1.2 – O que é abdução?**

Abdução é um tipo de raciocínio que tem como objetivo formular hipóteses para explicar fatos observados, considerando uma teoria como fundamento (Josephson e Josephson, 1994). Psicologicamente, a noção<sup>1</sup> de abdução está presente quando uma pessoa cria hipóteses para:

- Explicar o fato de que o motor de um automóvel não dá partida.
- Identificar suspeitos de um assassinato, considerando como fatos pistas da cena do crime.
- Diagnosticar uma doença tendo como fatos os sintomas apresentados por um paciente.

---

<sup>1</sup> Diferentes noções de abdução são encontradas na literatura, embora elas possuam uma essência comum. Oliveira (2016) discute e exemplifica diferenças entre noções de abdução comumente empregadas em estudos das áreas de Computação, Psicologia e Filosofia.

- Interpretar o que a expressão “Fogo!” significa, considerando um conjunto de fatos contextuais.
- Classificar um ser vivo (mamífero, inseto, réptil etc.) a partir da identificação de suas características morfológicas (vertebrado, invertebrado, endotérmico, ectotérmico etc.)

Em cada um destes exemplos<sup>2</sup> a teoria usada como fundamento pela pessoa que raciocina corresponde informalmente à noção de conhecimento que ela possui sobre mecânica de automóveis, *modus operandi* de criminosos, relações humanas, medicina, língua portuguesa, teoria da evolução, morfologia, entre tantos outros.

Diferentemente dos raciocínios dedutivos, em que existe conservação da verdade, as hipóteses criadas em um raciocínio abductivo são meras possibilidades, i.e., podem ou não ser verdade. No exemplo do motor de um automóvel que não dá partida, são possíveis hipóteses, entre outras, a bateria estar sem carga, estar faltando combustível e o motor de arranque estar danificado. Qualquer delas é uma possível hipótese, mas não há garantia de que alguma seja a verdadeira explicação para o fato do motor não dar partida.

### 1.3 – Objetivos deste trabalho

Abdução tem sido amplamente estudada no contexto da busca de soluções factíveis para muitos problemas incluindo diagnóstico, entendimento de linguagem natural, planejamento, formulação de argumentos, aprendizagem de máquina, revisão de crenças e raciocínios na Web Semântica (Gavanelli *et al.*, 2015). As soluções computacionais comumente propostas nestes estudos incluem o uso de *frameworks* que estendem a programação lógica tradicional com sistemas de programação lógica com restrições (*Constraint Logic Programming* – CLP) (Jaffar & Maher, 1994). Exemplos destas soluções incluem Programação Lógica Abductiva (*Abductive Logic Programming* – ALP) (Kakas, Kowalski & Toni, 1993), Programação Lógica Abductiva com Restrições (*Abductive Constraint Logic Programming* – ACLP) (Kakas, Michael & Mourlas, 2000)

---

<sup>2</sup> Rodrigues (2015) é uma fonte para uma grande diversidade de exemplos de raciocínios abductivos.

e Regras para Tratamento de Restrições (*Constraint Handling Rules* – CHR) (Abdennadher & Christiansen, 2000; Frühwirth, 1998).

Embora a programação de raciocínios abduativos possa ser usada em um contexto de modelagem educacional, nenhum estudo fora encontrado sobre este assunto. Isto motiva a proposta de uma nova modalidade de modelagem computacional educacional que se define pela programação de modelos para computação de raciocínios abduativos. Especificamente, este trabalho objetiva:

- O desenvolvimento de diretrizes para fundamentar, organizar e orientar as atividades do que, neste trabalho, está sendo denominado de modelagem computacional educacional abdutiva.
- Uma investigação experimental da atividade de modelagem computacional educacional abdutiva.

#### **1.4 – Organização deste documento**

O restante deste documento está organizado da seguinte maneira.

*Capítulo 2 – Modelagem Computacional Educacional:* Introduce a área de modelagem computacional educacional, dá exemplos de sistemas computacionais para modelagem e descreve o funcionamento típico de um ambiente educacional para modelagem computacional, situando os papéis do estudante e do professor.

*Capítulo 3 – Abdução:* Define raciocínio abduativo e ilustra como raciocínios abduativos podem ser formalizados usando Lógica de Primeira Ordem (LPO).

*Capítulo 4 – Metodologia:* Descreve a metodologia empregada no trabalho.

*Capítulo 5 – Proposta de diretrizes para Modelagem Computacional Educacional Abdutiva:* Estabelece a proposta autoral de diretrizes para fundamentar, organizar e orientar as atividades em um ambiente para modelagem computacional educacional abdutiva. As diretrizes definem como o ambiente para modelagem irá funcionar, o que o estudante deve fazer e como ele deve interpretar os resultados para tomar a decisão de reescrever ou não um modelo. As diretrizes propostas são gerais, não dependem de

nenhuma linguagem de programação abdutiva. Em particular, é dado um exemplo de uso das diretrizes empregando a Lógica de Primeira Ordem.

*Capítulo 6 – Linguagem CHR: introdução à programação e uso para abdução:* Apresenta uma introdução à linguagem *Constraint Handling Rules* (CHR) e mostra como ela pode ser utilizada para a programação de raciocínios abduativos. Esta linguagem é utilizada nos experimentos que foram conduzidos.

*Capítulo 7 – Pesquisa Experimental sobre a Atividade de Modelagem Computacional Educacional Abdutiva:* Usando a linguagem CHR como mecanismo expressivo, o capítulo apresenta o planejamento de uma pesquisa experimental sobre a atividade de modelagem computacional abdutiva. São discutidos os materiais, procedimento experimental e os sujeitos da pesquisa.

*Capítulo 8 – Discussão e Conclusão:* Discute e apresenta as conclusões deste trabalho.

*Apêndice I – Formulário: pesquisa experimental sobre modelagem computacional educacional abdutiva:* Descreve integralmente o Formulário planejado para assistir às sessões da pesquisa experimental.

*Apêndice II – Padrão de Resposta aos Exercícios da Pesquisa Experimental:* Descreve um padrão para as respostas consideradas correta aos exercícios que integram as sessões da pesquisa experimental.

*Apêndice III – Artigos publicados e em processo de avaliação:* São citados artigos já publicados e a serem publicados com os resultados deste trabalho.

# Capítulo 2

## Modelagem Computacional Educativa

*Este Capítulo detalha o que é Modelagem Computacional Educativa no contexto deste trabalho. A Seção 2.1 aborda a origem, a evolução e ilustra a construção de um modelo computacional, desenvolvido com propósito educativo. A Seção 2.2 discute a “anatomia” de um ambiente para modelagem computacional educativa típico. Em especial, os artefatos e os papéis desempenhados pelos atores envolvidos no ambiente são analisados e relatados.*

## 2.1 – Modelagem computacional educacional: origem e evolução

**Modelagem computacional** refere-se à atividade de criar programas que estabelecem modelos para fenômenos, i.e., simulam em computador fenômenos reais ou abstratos. No campo da Educação, a modelagem tem sido utilizada como uma opção didática para implementar um paradigma educacional construtivista (Papert, 1993), no qual o estudante é convidado a aprender através da atividade de construção de modelos. A ideia fundamental é a de ofertar ao estudante uma linguagem de programação e motivá-lo a se colocar em um ciclo no qual ele programa a sua concepção sobre um fenômeno (i.e., cria um modelo computacional sobre o fenômeno), executa o programa (modelo) desenvolvido, interpreta os resultados da execução e, se os resultados são inconsistentes com o fenômeno, reformula o modelo desenvolvido, à medida em que também reconstrói o seu entendimento sobre o fenômeno, aprendendo com isto.

Modelagem é mais do que a simples execução de simulações, entendendo-se por simulação a atividade de investigação de como a mera troca de parâmetros afeta as propriedades capturadas por modelos pré-programados. Modelagem envolve a criação de modelos e a confrontação da coerência conceitual do modelo com a realidade ou com o conhecimento estabelecido e aceito.

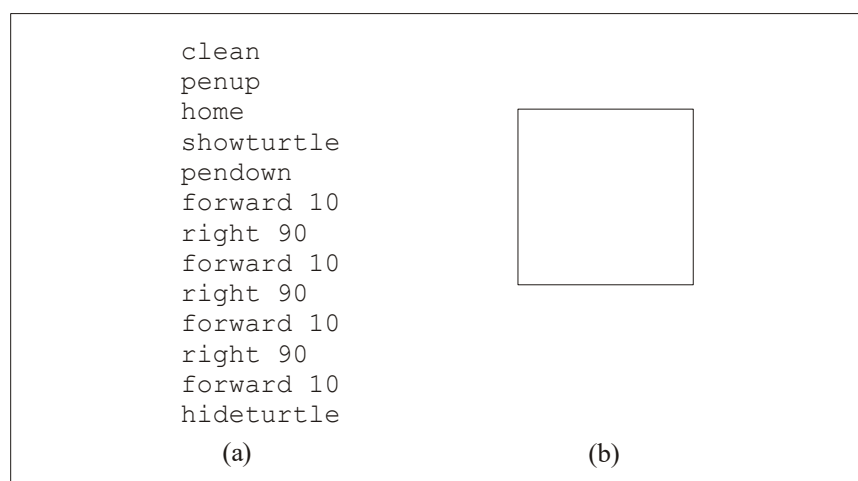
O desenvolvimento de linguagens de programação para modelagem com propósitos educacionais tem origem no final da década de 1960 em estudos envolvendo a linguagem LOGO (Papert, 1993).

**Exemplo 2.1 (Um modelo em linguagem LOGO).** LOGO, mais especificamente a funcionalidade da linguagem LOGO conhecida como LOGO Geométrico, utiliza o desenho de uma tartaruga sobre a tela de um computador como metáfora para movimento e desenho. A “tartaruga pode andar pela tela” e, na medida em que se movimenta, pode deixar uma linha como rastro por onde passa. A linguagem LOGO pode ser empregada para criar modelos para conceitos da geometria. Dirigida originalmente para crianças, a metáfora assumida pela linguagem, estabelece a ideia de comandar movimentos da tartaruga por meio de instruções (MSWLogo, 2017) como:

- `forward 10`: que faz a tartaruga andar 10 passos para a frente.

- `right 90`: que faz a tartaruga girar 90 graus à direita.
- `left 90`: que faz a tartaruga girar 90 graus à esquerda.
- `clean`: que limpa a tela deixando-a sem desenhos.
- `home`: que posiciona a tartaruga no centro da tela.
- `penup`: que instrui a tartaruga a não desenhar enquanto se move.
- `pendown`: que instrui a tartaruga a desenhar enquanto se move.
- `hideturtle`: que esconde a tartaruga.
- `showturtle`: que mostra a tartaruga na tela.

O programa em linguagem LOGO da Figura 2.1(a) estabelece o que estamos chamando de modelo. Especificamente, esta figura estabelece um modelo para uma instância do conceito geométrico de quadrado. A execução do modelo da Figura 2.1(a) é um desenho que, no caso descrito, está apresentado na Figura 2.1(b). O resultado da execução do modelo pelo computador cria para o modelador (estudante) a possibilidade dele confrontar o modelo estabelecido com o conceito de quadrado. Se o resultado não é coerente com um quadrado, por que ele não sabe exatamente o que é um quadrado, ou sabe e não o descreveu corretamente, ele pode reformular o modelo à medida em que também tem chance de reformular o seu entendimento do que é um quadrado (quadrilátero formado por quatro lados de mesmo comprimento e quatro ângulos retos).



**Figura 2.1. Um modelo desenvolvido na linguagem LOGO: (a) O modelo propriamente. (b) O resultado da execução do modelo.**

Após os estudos pioneiros de Papert sobre modelagem computacional, seguiram-se décadas de pesquisa sobre aspectos pedagógicos e epistemológicos (Campbell & Oh, 2015) da prática escolar da Aprendizagem baseada em Modelos (em Inglês, MbL – *Modeling-based Learning*). Paralelamente, no campo computacional, foram utilizadas para modelagem várias linguagens de propósito geral, predominantemente de paradigma imperativo, orientado a objetos e funcional (e.g., C, Java, Lisp). Também foram desenvolvidos sistemas de propósito específicos, entre os quais se destacam os que foram dedicados à modelagem de sistemas dinâmicos, i.e., sistemas que dependem do tempo. STELLA (*Structural Thinking Experimental Learning Laboratory with Animation*) (Richmond, Peterson & Vescuso, 1987) e IQON (*Interacting Quantities Omitting Numbers*) (Miller *et al.*, 1990) são dois exemplares destes sistemas, o primeiro dedicado à modelagem quantitativa e o segundo à modelagem semi-quantitativa de sistemas dinâmicos. Modelos em STELLA e IQON são grafos em que as variáveis dinâmicas são representadas como vértices e as arestas determinam o relacionamento entre elas. Matematicamente um modelo de STELLA ou de IQON representam sistemas de equações diferenciais. As variáveis dinâmicas que participam de um modelo em STELLA assumem valores do conjunto dos números reais, enquanto que as variáveis que participam de um modelo IQON assumem valores numa escala que pode variar entre o nada, o pouco, o médio, o muito e o total.

Exemplos de estudos recentes sobre modelagem no contexto educacional incluem investigações sobre o uso de sistemas como Scratch (Lopez & Hernandez, 2015), MatLab (Downey, 2011), e CellCollective (Helikar *et al.*, 2015). Scratch tem sido usado principalmente para modelagem de animações, Matlab para modelagem matemática (não simbólica) em física e engenharia e CellCollective para modelagem de processos biológicos.

## **2.2 – Ambiente para modelagem computacional típico, atores e papéis desempenhados**

Um ambiente para modelagem computacional é composto tipicamente pelos seguintes artefatos e pessoas:

- Computador.
- Linguagem de programação e software para execução de programas.
- Estudantes.
- Professor.

Modelagem pressupõe um processo no qual modelos são construídos gradativamente em um ciclo que envolve a escrita (programação), a execução computacional, a avaliação de resultados e a possível modificação de um modelo. Neste processo, a aprendizagem se dá pela atividade do estudante continuamente construir e reconstruir modelos, uma vez que ele necessita buscar conhecimentos sobre o que está sendo modelado, expressar estes conhecimentos via linguagem de programação e avaliar o resultado da execução. A presença do computador é fundamental neste ambiente porque produz os resultados do que foi programado e, o que foi programado corresponde ao conhecimento do estudante. O estudante, com isto, pode avaliar o seu próprio conhecimento.

A educação em um ambiente para modelagem se processa predominantemente pelas vias da aprendizagem e não pelas vias do ensino. A rigor, o estudante não precisa ser ensinado, o processo o leva a aprender à medida em que constrói e reconstrói um modelo.

Então quais são os papéis dos estudantes e professores neste ambiente?

O estudante não desempenha o papel de um aluno que, passivamente, recebe informações transmitidas por um professor, em uma perspectiva pedagógica tradicional instrucionista. Ele é um agente ativo que constrói o seu próprio conhecimento pela atividade de modelagem, em uma perspectiva pedagógica construtivista ou, usando um termo, com origem em Papert (1993), “construcionista”. O papel do estudante é, portanto, de um aprendiz, termo este que é consagrado na literatura “construcionista”.

O professor não é o agente que transmite diretamente conhecimentos, posto que a atividade de modelagem não requer ensinar o aprendiz. Ele deve ser um mediador das experiências de modelagem do aprendiz, atuando para facilitar as experiências de modelagem do aprendiz por meio de ações que motivem, desafiem e orientem o aprendiz. O professor (mediador) deve atuar em uma zona de ação que promove a continuidade do ciclo de modelagem para que o aprendiz não desista da atividade de

modelagem, seja porque não se sinta desafiado ou não consiga ultrapassar as dificuldades relacionadas com o estabelecimento de um modelo.

# Capítulo 3

## Abdução

*O conceito de abdução (Seção 1.2), um tipo de raciocínio, é central a este trabalho, uma vez que é este o conceito que qualifica a classe de modelagem computacional educacional aqui tratada. O capítulo inicia com a apresentação de uma definição formal, que captura a essência das variadas definições de raciocínio abduutivo encontradas na literatura. A definição apresentada, de vertente lógica, usa a Lógica de Primeira Ordem (LPO) como formalismo para ser expressa. A Seção 3.2 apresenta um exemplo sobre como usar o formalismo tratado nas seções anteriores para descrever um sistema para raciocínio abduutivo. O capítulo termina com uma discussão, de caráter extra-lógico (Seção 3.3), sobre critérios para escolha de boas hipóteses entre várias possíveis hipóteses inferidas em um raciocínio abduutivo.*

### 3.1 – Uma definição formal de raciocínio abduutivo

Este trabalho utiliza a Lógica de Primeira Ordem (LPO) como fundamento para descrição de raciocínios abdutivos. Os conceitos de variáveis, constantes, funções, predicados (fórmula atômica), conectivos lógicos, quantificadores, termo, funtor, aridade, termo composto, fórmula bem-formada, interpretação, modelo, substituição, tautologia, contradição, consequência lógica, equivalência lógica e resolução são definidos de maneira usual e não são rediscutidos neste texto. Nicoletti (2017) e Pospel (2003) são duas boas fontes para estes conceitos.

Diferentes definições formais para raciocínio abduutivo em lógica têm sido propostas, mas de maneira geral, em essência elas compartilham o que está declarado na Definição 3.1.

**Definição 3.1 (Uma definição de raciocínio abduutivo):** dado um sistema  $\langle T, F \rangle$  onde:

- $T$ : descreve uma teoria por meio de um conjunto finito e não vazio,  $T = \{ t_1, t_2, t_3, \dots, t_m \}$ , de sentenças pertencentes à LPO, denotando  $t_1 \wedge t_2 \wedge t_3 \wedge \dots \wedge t_m$ , e
- $F$ : descreve um conjunto finito e não vazio,  $F = \{ f_1, f_2, f_3, \dots, f_q \}$ , de fatos<sup>3</sup> que devem ser explicados pelo raciocínio abduutivo, denotando  $f_1 \wedge f_2 \wedge f_3 \wedge \dots \wedge f_q$ , em que cada  $f_i (1 \leq i \leq q)$  é um predicado pertencente à LPO.

Realizar um raciocínio abduutivo sobre o sistema  $\langle T, F \rangle$  é calcular um conjunto de hipóteses  $H$  satisfazendo:

$$T \not\models F, \quad (3.1)$$

$$T \cup \{ h \} \models F, \forall h \in H, \quad (3.2)$$

$$T \cup \{ h \} \not\models \perp, \forall h \in H. \quad (3.3)$$

$H = \{ h_1, h_2, h_3, \dots, h_p \}$  deve ser lido como  $h_1 \vee h_2 \vee h_3 \vee \dots \vee h_p$ .

■

A condição (3.1) garante que  $T$  é geral, i.e., não explica diretamente (sem a necessidade de hipóteses) fatos presentes no conjunto  $F$ . Hipóteses  $h \in H$  satisfazendo à condição

---

<sup>3</sup> Fato é termo genérico que está sendo usado para referenciar evidências, sintomas, observações, marcas, sinais etc..

(3.2) são chamadas *hipóteses candidatas*, e explicam os fatos presentes no conjunto  $F$ . Hipóteses candidatas satisfazendo à condição (3.3) são *hipóteses consistentes*, i.e., hipóteses que quando empregadas em conjunção com  $T$ , não geram uma contradição.

### 3.2 – Exemplo de formalização de raciocínios abduativos

O Exemplo 3.1, a seguir apresentado, emprega a Definição 3.1 para formalizar raciocínios abduativos sobre uma pequeníssima parte da conhecida Classificação Biológica dos Seres Vivos (Wikipédia, 2017).

**Exemplo 3.1 (Uma instanciação da Definição 3.1).** Considere o conjunto teoria  $T_1$  e os conjuntos de fatos  $F_1$ ,  $F_2$  e  $F_3$  onde:

$$T_1 = \{$$

$$\quad \text{mamífero}(X) \rightarrow \text{vertebrado}(X) \wedge \text{endotérmico}(X),$$

$$\quad \text{réptil}(X) \rightarrow \text{vertebrado}(X) \wedge \text{ectotérmico}(X)$$

$$\}.$$

$$F_1 = \{ \text{vertebrado}(\text{cachorro}), \text{endotérmico}(\text{cachorro}) \}.$$

$$F_2 = \{ \text{ectotérmico}(\text{serpente}) \}.$$

$$F_3 = \{ \text{vertebrado}(\text{crocodilo}) \}.$$

O conjunto  $T_1$  é um pequeno modelo para classificar os seres vivos. A teoria  $T_1$  descreve sentenças da LPO, com variáveis e constantes, que tentam capturar o que, em português poderia ser aproximadamente descrito como: (1) se  $X$  é um mamífero então  $X$  é vertebrado e endotérmico, i.e., possui um sistema que regula a temperatura corporal; (2) se  $X$  um é réptil então  $X$  é vertebrado e ectotérmico, i.e., não possui sistema que mantém a temperatura corporal constante. Os conjuntos de fatos  $F_1$ ,  $F_2$  e  $F_3$  descrevem características morfológicas observadas em seres vivos por meio de predicados da LPO.

Possíveis resultados dos raciocínios abduativos descrito por:

- $\langle T_1, F_1 \rangle$  é  $H_1 = \{ \text{mamífero}(\text{cachorro}) \}$ .
- $\langle T_1, F_2 \rangle$  é  $H_2 = \{ \text{réptil}(\text{serpente}) \}$ .

- $\langle T_1, F_3 \rangle$  é  
 $H_3 = \{ \text{mamífero}(\text{crocodilo}), \text{réptil}(\text{crocodilo}), \text{mamífero}(\text{crocodilo}) \wedge \text{réptil}(\text{crocodilo}) \}$ .

Os conjuntos de hipóteses  $H_1$ ,  $H_2$  e  $H_3$  satisfazem, em cada um dos três raciocínios, às condições (3.1), (3.2) e (3.3) da Definição 3.1 e, por isto, são possíveis soluções em cada caso. Nos dois primeiros raciocínios, os fatos apresentados foram tais que, em cada caso, produzem uma única hipótese como resposta. Diferentemente, o terceiro raciocínio produz três hipóteses no conjunto  $H_3$ : elas devem ser interpretadas como “*mamífero(crocodilo)*” ou “*réptil(crocodilo)*” ou, conjuntamente, “*mamífero(crocodilo)  $\wedge$  réptil(crocodilo)*”. Ou seja, considerando a teoria  $T_1$  e o conjunto de fatos  $F_3$ , as três hipóteses são plausíveis. Neste caso, isto ocorreu por que o conjunto de fatos apresentados não é suficiente, em relação à teoria  $T_1$ , para determinar uma ou outra hipótese. Além disso, a hipótese “*mamífero(crocodilo)  $\wedge$  réptil(crocodilo)*” é especial no seguinte sentido: se considerarmos que, na Classificação Biológica, as classes de mamíferos e répteis são mutuamente exclusivas então pode-se dizer que a teoria  $T_1$  e a Classificação Biológica são discordantes. Se o objetivo é fazer com que haja concordância entre a teoria  $T_1$  e a Classificação Biológica, então a teoria  $T_1$  deveria ser modificada para refletir tal concordância.

Acrescentando mais reflexões a esta discussão:

- A condição (3.1) estabelece que não há solução (conjunto de hipóteses) quando um conjunto teoria  $T$  explica diretamente um conjunto de fatos  $F$ . Por exemplo, o raciocínio definido por  $\langle T_2, F_5 \rangle$ , onde

$$T_2 = \{ \text{mamífero}(\text{cachorro}), \text{réptil}(\text{serpente}) \} \text{ e}$$

$$F_5 = \{ \text{mamífero}(\text{cachorro}) \},$$

não possui solução por que  $T_2 \vDash F_5$  e isto viola a condição (3.1).

- A condição (3.2) estabelece que

$$h = \text{réptil}(\text{cachorro})$$

não é solução para o raciocínio abduutivo descrito por  $\langle T_1, F_1 \rangle$  por que

$$T_1 \cup \{ \text{réptil}(\text{cachorro}) \} \not\vDash F_1.$$

- A condição (3.3) estabelece que

$$h = \text{mamífero}(\text{cachorro}) \wedge \neg \text{endotérmico}(\text{cachorro})$$

não é solução para o raciocínio abduutivo descrito por  $\langle T_1, F_1 \rangle$  por que

$$T_1 \cup \{ \text{mamífero}(\text{cachorro}) \wedge \neg \text{endotérmico}(\text{cachorro}) \} \models \perp.$$

■

### 3.3 – Critérios para escolha de boas hipóteses

O conjunto  $H$  resultante de um raciocínio abduutivo contém possíveis hipóteses para explicar um conjunto de fatos  $F$ , tendo por fundamento uma teoria  $T$ . Estudos filosóficos (e.g., Thagard (1989) e Thagard (1978)), psicológicos (e.g., Magnani, (2009)) e computacionais (e.g., Caroprese & Trubitsyna, 2014) sugerem que algumas hipóteses podem explicar melhor um conjunto de fatos do que outras. Assim, definições de raciocínio abduutivo adotadas em certos estudos, não é o caso da Definição 3.1, declaram critérios adicionais para escolha de boas hipóteses de tal modo que afirmam que o resultado de um raciocínio abduutivo é um conjunto de hipóteses  $B \subseteq H$ , onde  $B$  contém somente hipóteses consideradas boas. Por exemplo, Rodrigues, Oliveira e Oliveira (2014) criam tais definições adicionais, propondo dois critérios, usados em conjunto, para classificar uma hipótese como boa: “uma hipótese é boa se ela possuir mínima complexidade entre todas aquelas que possuem máxima força explanatória”. Heurísticas são definidas para estabelecer valores numéricos para a complexidade e a força explanatória de uma hipótese. A complexidade<sup>4</sup> de uma hipótese é definida como a quantidade de predicados<sup>5</sup> que ela possui. A força explanatória<sup>6</sup> de uma hipótese é definida como a relação entre a quantidade de fatos que ela explica e quantidade total de fatos a serem explicados.

■

<sup>4</sup> O conceito de simplicidade (ou, inversamente, complexidade) proposto por muitas abordagens deriva do princípio de Occam (Russel & Norvig, 2014).

<sup>5</sup> Trabalhando em Lógica Proposicional, Rodrigues, Oliveira e Oliveira (2014) referem-se a “quantidade de literais”. Este trabalho está tomando a liberdade de adaptar a ideia para a LPO, quando descreve o conceito de complexidade a partir de “quantidade de predicados”.

<sup>6</sup> Rodrigues, Oliveira e Oliveira (2014) usam uma definição de raciocínio abduutivo que permite que uma hipótese explique parcialmente o conjunto de fatos a serem explicados pelo raciocínio abduutivo.

**Exemplo 3.2 (Critérios adicionais para seleção de boas hipóteses).** O raciocínio definido por  $\langle T_1, F_3 \rangle$  do Exemplo 3.1 tem como solução o conjunto de hipóteses

$$H_3 = \{ \text{mamífero}(\text{crocodilo}), \text{réptil}(\text{crocodilo}), \\ \text{mamífero}(\text{crocodilo}) \wedge \text{réptil}(\text{crocodilo}) \}.$$

Não há um consenso sobre que critérios devem ser usados para escolha de boas hipóteses<sup>7</sup>. Este exemplo usa os critérios propostos por Rodrigues, Oliveira e Oliveira (2014). A força explanatória das três hipóteses presentes em  $H_3$  é igual a 1, pois cada uma destas hipóteses explica totalmente os fatos do conjunto  $F_3$ . As complexidades das hipóteses  $\text{mamífero}(\text{crocodilo})$ ,  $\text{réptil}(\text{crocodilo})$  e  $\text{mamífero}(\text{crocodilo}) \wedge \text{réptil}(\text{crocodilo})$  são iguais, respectivamente, a 1, 1 e 2, que são os valores correspondentes à quantidade de predicados presentes em cada uma delas. Assim, as hipóteses do conjunto

$$B = \{ \text{mamífero}(\text{crocodilo}), \text{réptil}(\text{crocodilo}) \}$$

são consideradas boas porque possuem complexidade mínima entre as hipóteses de  $H_3$  que exibem força explanatória máxima. A hipótese

$$B = \{ \text{mamífero}(\text{crocodilo}) \wedge \text{réptil}(\text{crocodilo}) \}$$

não é considerada boa. ■

---

<sup>7</sup> Apenas para citar a diversidade de critérios propostos, além dos critérios propostos por Rodrigues, Oliveira e Oliveira (2014), Caroprese e Trubitsyna (2014) propõem o conceito de *arbitrariedade* de uma hipótese, e afirmam ser preferível escolher as hipóteses menos arbitrárias. Thagard (1989) propõe o conceito de *coerência explanatória* de uma hipótese, e afirma ser preferível escolher as mais coerentes com a teoria utilizada.

# Capítulo 4

## Metodologia

*Este capítulo descreve a metodologia empregada neste trabalho e é composto por duas seções: a primeira (Seção 4.1) oferece uma visão geral da metodologia e a segunda (Seção 4.2) detalha cada uma das etapas metodológicas.*

## 4.1 – Visão geral da metodologia

A metodologia empregada é composta pelas seguintes etapas:

- Revisão bibliográfica;
- Estudos de linguagens e práticas de programação;
- Estudos sobre Lógica de Primeira Ordem;
- Proposta de diretrizes para modelagem computacional educacional abdutiva;
- Proposta de um ambiente para modelagem computacional abdutiva;
- Planejamento da pesquisa experimental sobre atividade de modelagem computacional educacional abdutiva;
- Realização da pesquisa experimental;
- Análise quantitativa e qualitativa dos resultados da pesquisa experimental.

A Figura 4.1 oferece uma visão pictórica da metodologia empregada.



Figura 4.1. Visão pictórica da metodologia empregada.

## 4.2 – Descrição das etapas metodológicas

*Revisão bibliográfica:* a revisão abrangeu os temas de modelagem computacional, computação lógica de raciocínio abduutivo e linguagem CHR, por meio de acesso às bibliotecas digitais da ACM e IEEE, *White Papers* livremente disponíveis na Internet, e livros disponíveis em papel ou em formato digital. O método de revisão utilizado, algumas vezes denominado de revisão “narrativa”, não contou com técnicas de revisão sistemática. A revisão não seguiu um protocolo rígido para sua realização, como é o caso das revisões sistemáticas. A seleção dos artigos foi realizada seguindo referências clássicas em cada tema de interesse e referências para as referências clássicas, a partir de buscas por palavras chave nas bibliotecas digitais e buscas livres na Internet. Referências das referências, subjetivamente avaliadas como importantes para os objetivos deste trabalho, também são utilizadas.

*Estudos de linguagens e práticas de programação:* foram realizados estudos e práticas na programação em linguagem Prolog (Bramer, 1993), programação em linguagem CHR (Frühwirth, 2009) e operação do ambiente SWI-Prolog (SWI-Prolog-Org, 2017). O método utilizado foi o de estudar as referências apontadas e praticar em computador a programação em linguagem Prolog, linguagem CHR dentro do ambiente SWI-Prolog.

*Estudos sobre Lógica de Primeira Ordem:* foram realizados estudos sobre a Lógica de Primeira Ordem (Nicoletti, 2017; Pospesel, 2003) com o objetivo de definir diretrizes para Modelagem Computacional Educacional Abduativa de maneira independente de uma linguagem de programação específica.

*Proposta das diretrizes para modelagem computacional educacional abduativa:* a proposta destas diretrizes teve como base metodológica o funcionamento de ambientes para modelagem, descritos na bibliografia consultada, e a prática adquirida na programação de raciocínios abduativos tanto em linguagem Abdl (Oliveira, Oliveira, Martins & Matsumoto, 2016), sistema Peirce Online (Rodrigues, Oliveira & Oliveira, 2014) e linguagem CHR (Frühwirth, 2009). Em especial, destaca-se que a experiência prática de programação abduativa contribuiu para acrescentar às diretrizes o conceito de intenção de modelagem, as distintas situações podem resultar da avaliação de um

conjunto de hipóteses e a decisão pela continuidade ou não do ciclo de modelagem (itens 4 e 5 da Definição 5.1).

*Proposta de um ambiente para modelagem computacional abdutiva:* Um ambiente para modelagem computacional abdutiva tendo como componentes e atores estações de trabalho equipadas com sistema SWI-Prolog (SWI-Prolog-Org, 2017), estudantes e um mediador.

*Planejamento da pesquisa experimental sobre atividade de modelagem computacional educacional abdutiva:* esta pesquisa experimental teve como principal fundamento metodológico as diretrizes propostas. As habilidades que foram propostas para serem investigadas seguem diretamente das diretrizes propostas. O detalhamento da metodologia da pesquisa experimental está feito na Seção 7.1.

*Realização da pesquisa experimental:* o preparo, a execução e a apuração de resultados seguiram recomendações metodológicas tradicionais para condução de sessões de pesquisa experimental. As recomendações metodológicas para pesquisa experimental de Lazar, Feng & Hochheiser (2010) foram particularmente úteis. O detalhamento da metodologia para análise quantitativa e qualitativa dos resultados está feito na Seção 7.5.

*Análise quantitativa e qualitativa dos resultados da pesquisa experimental:* os resultados quantitativos da pesquisa experimental foram analisados usando Análise de Variância (ANOVA) e testes *t* de *Student*. A análise qualitativa baseou-se em uma avaliação empírica sobre as anotações realizadas pelo pesquisador nas sessões experimentais. A metodologia utilizada nas análises está detalhada na Seção 7.6.

# Capítulo 5

## Proposta de diretrizes para Modelagem Computacional Educativa

*Este capítulo descreve as diretrizes para a modelagem computacional educativa, proposto neste trabalho. A Seção 5.1 declara especificamente, para este trabalho, o que significam tais diretrizes. A Seção 5.2 apresenta as diretrizes e chama a atenção para algumas de suas qualidades. Por fim, a Seção 5.3 exemplifica um uso das diretrizes propostas.*

## **5.1 – O que são as diretrizes no contexto deste trabalho?**

O termo diretrizes, de que trata este capítulo, está empregado em um sentido comumente utilizado na área de educação, para discriminar os conceitos, procedimentos e decisões que ocorrem em um ambiente educacional que emprega a modelagem computacional abdutiva, atuando como um guia ordenador e orientador da atividade que ocorre dentro de um ambiente de modelagem computacional de raciocínios abduativos, com propósitos educacionais.

## **5.2 – Diretrizes para modelagem computacional educacional abdutiva**

Ambientes para modelagem computacional educacional têm em comum o ciclo no qual o estudante escreve, executa, avalia e, possivelmente, reescreve um modelo. Todavia, estes ambientes são diferentes nos conceitos e paradigmas de programação que empregam. Por exemplo, ambientes clássicos para modelagem de desenhos de geometria plana, como os descritos por Papert (1993), empregam conceitos de geometria plana e usam uma linguagem declarativa, de lógica dedutiva. Diferentemente, ambientes para modelagem computacional abdutiva utilizam conceitos de teoria e fatos numa lógica não dedutiva, para produzir conjuntos de possíveis hipóteses.

As diretrizes consubstanciam a maneira como o ambiente para modelagem irá funcionar, o que o estudante deve fazer e como ele deve interpretar os resultados para tomar a decisão de reescrever ou não um modelo.

Especificamente, as diretrizes para modelagem computacional educacional abdutiva propostas neste trabalho:

- Utilizam os conceitos de teoria, fato e hipótese, conforme a Definição 3.1.
- Declaram que um modelo deve ser descrito pelo conceito de teoria.
- Declaram que o estudante deve propor fatos para elaborar raciocínios abduativos que, executados pelo computador, resultam em hipóteses para serem avaliadas.
- Discriminam, como se verá, quatro situações distintas que podem surgir da avaliação das hipóteses resultantes da execução de raciocínios abduativos.

- Para cada uma das quatro situações que podem surgir da avaliação de hipóteses, oferecem orientações ao estudante sobre a decisão que deve tomar em relação a encerrar o trabalho de modelagem ou reescrever um modelo afim de aperfeiçoá-lo.

Enfim, as diretrizes para modelagem computacional educacional abdutiva propostas estão declaradas por meio da Definição 5.1, dada a seguir.

**Definição 5.1 (Diretrizes para modelagem computacional educacional abdutiva).** As diretrizes para modelagem computacional educacional abdutiva compreendem uma atividade de modelagem em que o estudante se insere em um ciclo no qual:

1. Formula uma teoria lógica  $T$  sobre um fenômeno.
2. Levanta fatos  $F_i$  ( $i \geq 1$ ) pertinentes ao fenômeno.
3. Executa em computador os raciocínios abdutivos descritos por  $\langle T, F_i \rangle$  ( $i \geq 1$ ) para obter respectivos conjuntos de hipóteses  $H_i$ .
4. Avalia o conjunto de hipóteses  $H_i$  ( $i \geq 1$ ) em relação à intenção de modelagem. Intenções típicas de modelagem compreendem o desenvolvimento de modelos que (1) imitam a realidade, (2) sejam coerentes com alguma teoria científica existente ou conhecimento consolidado, (3) sejam alternativas a teorias existentes, (4) sejam surreais por opção etc.. Quatro distintas situações podem resultar da avaliação de um conjunto de hipóteses  $H_i$ :
  - 4.1. Todas as hipóteses  $h \in H_i$  são coerentes com a intenção de modelagem.
  - 4.2.  $H_i = F_i$ , onde  $F_i$  é o conjunto de fatos apresentados em um raciocínio  $H_i$ .  
Nesta situação, as hipóteses em  $H_i$  não são explanatórios, ou seja, as hipóteses não explicam os fatos, através das teorias e condições utilizadas em um raciocínio  $H_i$ .
  - 4.3. Existe pelo menos uma hipótese  $h \in H_i$  não coerente com a intenção de modelagem.
  - 4.4. O conjunto  $H_i = \emptyset$ . Neste caso, não existem hipóteses que explicam o conjunto de fatos presentes em um raciocínio  $H_i$ , ou seja, a teoria  $T$  é insuficiente para explicar os fatos.

5. Decide alterar teoria  $T$  e volta ao passo 2 ou aceita a teoria desenvolvida e dá o trabalho de modelagem por encerrado. A decisão de alteração ou não da teoria depende da avaliação feita no passo 4. A situação 4.1 normalmente é um indicativo de que a teoria desenvolvida é coerente com a intenção da modelagem, mas não é uma garantia disto porque o modelador pode, à medida que se aprofunda no fenômeno que está sendo modelado, descobrir fatos que geram conjuntos de hipóteses que são incoerentes com a intenção de modelagem. A própria intenção de modelagem pode ser alterada pelo modelador à medida que ele ganha conhecimento sobre o fenômeno que está sendo modelado. Na maioria das vezes, pode-se afirmar que a situação 4.1 motiva uma decisão pelo encerramento do trabalho de modelagem. Em contrapartida, as situações 4.2, 4.3 e 4.4 motivam a decisão pela continuidade do trabalho de modelagem, a primeira porque as hipóteses calculadas não explicam o fenômeno, através das teorias e condições utilizadas, a segunda porque a teoria  $T$  é incoerente com o fenômeno e a terceira porque a teoria  $T$  se mostra incompleta em relação a ele, considerando a intenção de modelagem.

■

Este ciclo de modelagem possibilita a um estudante aprender sobre o fenômeno que está sendo modelado porque ele entra em contato com os conceitos envolvidos no fenômeno, formaliza relações lógicas entre os conceitos, avalia o modelo desenvolvido em relação às intenções de modelagem e (re)cria o seu próprio conhecimento. A presença do computador como autômato que executa o modelo é fundamental para permitir que o estudante tenha um “outro” que, algumas vezes, lhe confirma o seu “modelo-entendimento” e também, outras vezes, o faça se surpreender com a imprecisão do seu conhecimento.

As quatro situações discriminadas em (4) e as orientações declaradas em (5) merecem destaque, porque configuram um apoio ao estudante, raramente presente em outros ambientes para modelagem, à importante questão do encerramento ou aperfeiçoamento de um modelo. Ter criado este apoio foi uma surpresa positiva, algo impossível de

imaginar quando se decidiu, no início deste trabalho, propor o que está sendo chamado de modelagem computacional educacional abdutiva.

As diretrizes propostas não incluem critérios para seleção de boas hipóteses, como os que são descritos na Seção 3.3. As razões da não inclusão de critérios para seleção de boas hipóteses se deve à decisão por diretrizes:

- Puramente lógicas, sem interferências extralógicas.
- Mais simples, focadas apenas nas hipóteses que resultam do cálculo lógico dos raciocínios modelados.

### 5.3 – Exemplo de uso das diretrizes propostas

O Exemplo 5.1 revisita o assunto da classificação biológica de seres vivos, tratada também em exemplos no Capítulo 3, para ilustrar o uso das diretrizes propostas.

**Exemplo 5.1 (Exemplo de uso das diretrizes propostas):** Considere como intenção de modelagem o desenvolvimento de um modelo coerente com uma pequeníssima parte, apresentada pela teoria  $T_1$ , da Classificação Biológica dos Seres Vivos (Wikipédia, 2017).

$$T_1 = \{ \\ \textit{mamífero}(X) \rightarrow \textit{vertebrado}(X) \wedge \textit{endotérmico}(X), \\ \textit{réptil}(X) \rightarrow \textit{vertebrado}(X) \wedge \textit{ectotérmico}(X) \\ \}.$$

Em seguida, após levantar conjuntos de fatos, pode-se executar raciocínios abdutivos como os seguintes:

- $\langle T_1, \{ \textit{vertebrado}(\textit{cachorro}), \textit{endotérmico}(\textit{cachorro}) \} \rangle$ , cuja resposta é  $H_1 = \{ \textit{mamífero}(\textit{cachorro}) \}$ .
- $\langle T_1, \{ \textit{vertebrado}(\textit{crocodilo}) \} \rangle$ , cuja resposta é  $H_2 = \{ \textit{mamífero}(\textit{crocodilo}), \textit{réptil}(\textit{crocodilo}), \textit{mamífero}(\textit{crocodilo}) \wedge \textit{réptil}(\textit{crocodilo}) \}$ .

A avaliação destes resultados é a seguinte. A hipótese contida em  $H_1$  é claramente coerente com a Classificação Biológica. Considerando o fato  $\{ \text{vertebrado}(\text{crocodilo}) \}$  do segundo raciocínio, também se pode afirmar que as hipóteses  $\text{mamífero}(\text{crocodilo})$  ou  $\text{réptil}(\text{crocodilo})$  são coerentes com a Classificação Biológica, embora tal classificação não considere um crocodilo como mamífero. Pode-se afirmar que há coerência entre o modelo da teoria  $T_1$  e a Classificação Biológica porque o raciocínio foi realizado com um conjunto insuficiente de fatos: na presença apenas do fato de que um certo ser vivo é vertebrado, tanto a teoria  $T_1$  quanto a Classificação Biológica permitem abduzir que tal ser vivo pode ser um mamífero ou um réptil.

A coerência dos resultados anteriores não motiva modificar a teoria  $T_1$ , mas não se pode dizer o mesmo da hipótese  $\text{mamífero}(\text{crocodilo}) \wedge \text{réptil}(\text{crocodilo})$  do conjunto  $H_2$ . Existe, neste caso, uma incoerência entre a teoria  $T_1$  e a Classificação Biológica: enquanto que para  $T_1$  algo pode ser classificado simultaneamente como mamífero e réptil, o mesmo não acontece com a Classificação Biológica. Assim, pode-se propor uma alteração da teoria  $T_1$  de modo que chega-se à teoria  $T_2$ , assim descrita:

$$T_2 = \{$$

$$\text{mamífero}(X) \rightarrow \text{vertebrado}(X) \wedge \text{endotérmico}(X),$$

$$\text{réptil}(X) \rightarrow \text{vertebrado}(X) \wedge \text{ectotérmico}(X),$$

$$\text{mamífero}(X) \rightarrow \neg \text{réptil}(X)$$

$$\}.$$

A alteração proposta captura a noção de que algo que é um mamífero não é um réptil por meio da inclusão da sentença  $\text{mamífero}(X) \rightarrow \neg \text{réptil}(X)$  à teoria  $T_1$ .

O ciclo se repete com a proposta de fatos e a execução de raciocínios abduzitivos usando agora a teoria  $T_2$ :

- $\langle T_2, \{ \text{vertebrado}(\text{crocodilo}) \} \rangle$ , cuja resposta é  $H_3 = \{ \text{mamífero}(\text{crocodilo}), \text{réptil}(\text{crocodilo}) \}$ .
- $\langle T_2, \{ \text{mamífero}(\text{rato-toupeira-pelado}), \text{ectotérmico}(\text{rato-toupeira-pelado}) \} \rangle$ , cuja resposta é  $H_4 = \emptyset$ .

As hipóteses presentes em  $H_3$  confirmam que a alteração de  $T_1$  para  $T_2$  resolveu o problema de inconsistência entre a Classificação Biológica e  $T_1$ . No entanto  $H_4$  é surpreendente no sentido de que revela que a teoria  $T_2$  é incapaz de produzir explicações para um ser vivo que é mamífero e ectotérmico. Este novo conjunto de fatos tem a ver com uma famosa exceção presente na Classificação Biológica: ratos-toupeira-pelado são mamíferos e não são endotérmicos. Esta situação também motiva alteração na teoria  $T_2$  para produzir uma nova teoria  $T_3$  e, novamente, o ciclo se repete:

$$T_3 = \{$$

$$\begin{aligned} & \textit{mamífero}(X) \rightarrow \textit{vertebrado}(X) \wedge \textit{endotérmico}(X), \\ & \textit{réptil}(X) \rightarrow \textit{vertebrado}(X) \wedge \textit{ectotérmico}(X), \\ & \textit{mamífero}(X) \rightarrow \neg \textit{réptil}(X), \\ & \textit{mamífero}(\textit{rato-toupeira-pelado}) \rightarrow \textit{vertebrado}(\textit{rato-toupeira-pelado}) \wedge \\ & \qquad \qquad \qquad \textit{ectotérmico}(\textit{rato-toupeira-pelado}) \end{aligned}$$

$$\}$$

■

## Capítulo 6

# Linguagem CHR: introdução à programação e uso para abdução

*Este Capítulo descreve a linguagem CHR (Constraint Handling Rules), concebida originalmente para a programação de solucionadores de restrições (constraint solvers) e, neste trabalho, usada para a programação de raciocínio abduativo. A Seção 6.1 descreve a origem, o propósito e como a linguagem CHR é usualmente implementada. Programas CHR são constituídos por três tipos de regras, as quais são discutidas na Seção 6.2. A Seção 6.3 apresenta, por meio de uma descrição informal e com exemplos, a semântica operacional de programas CHR. Por fim, a Seção 6.4 mostra como a linguagem CHR pode ser utilizada para desenvolver programas para computar raciocínio abduativo.*

## 6.1 – Introdução

A linguagem CHR (*Constraint Handling Rules*), desenvolvida na década de 1990, foi apresentada como um protótipo em Frühwirth (1998). Atualmente existem diversas implementações desta linguagem, sendo que, na maioria delas, ela não é apresentada como uma linguagem isolada, mas sim como uma extensão que se mistura com a sintaxe de uma linguagem hospedeira como, por exemplo, Prolog, Lisp, Haskell, C ou Java (Frühwirth, 2008). Este trabalho emprega uma implementação de CHR que é uma extensão da linguagem Prolog, disponível no ambiente SWI-Prolog (SWI-Prolog-Org, 2017). Sendo uma extensão linguística de Prolog, a implementação utilizada herda a nomenclatura básica da linguagem Prolog. Assim, aplicam-se à linguagem CHR as mesmas noções e definições atribuídas, pela literatura de Prolog — e.g., Bramer (2013) —, a constantes, variáveis, átomos, termos, predicados, aridade de predicado, regra, cláusula, cabeça e corpo de cláusula, questão, objetivo, uso de letras maiúsculas para variáveis etc.. De forma geral, programas podem ser escritos como uma combinação de sentenças das linguagens CHR e Prolog, embora este trabalho tem foco apenas nas sentenças da linguagem CHR.

A linguagem CHR foi concebida como uma linguagem especificamente destinada ao desenvolvimento de solucionadores de restrições (*constraint solvers*). A ideia de usá-la para implementar abdução foi proposta pela primeira vez em Abdennadher & Christiansen (2000) e, a partir daí, ela tem sido empregada para resolver inúmeros problemas que envolvem raciocínio abduutivo no planejamento, no diagnóstico médico, no processamento de linguagem natural, entre outros. Este trabalho propõe o uso da linguagem CHR como mecanismo para descrição de modelos dentro das diretrizes apresentadas na Definição 5.1. A escolha da linguagem CHR, em vez de outras opções disponíveis, tal como ALP (Kakas, Kowalski & Toni, 1993), ACLP (Kakas, Michael & Mourlas, 2000), Peirce Online (Rodrigues, Oliveira & Oliveira, 2014) e Abd1 (Oliveira, Oliveira, Martins & Matsumoto, 2016), se deve ao amplo uso, robustez e disponibilidade da linguagem CHR em relação às opções citadas e, também, ao propósito do autor de estudar esta linguagem.

## 6.2 – Regras CHR

Um programa CHR é composto por três tipos regras cujos formatos são:

- Regras de simplificação:  $h_1, h_2, \dots, h_n \Leftrightarrow \text{Guarda} \mid b_1, b_2, \dots, b_m$ .
- Regras de propagação:  $h_1, h_2, \dots, h_n \Rightarrow \text{Guarda} \mid b_1, b_2, \dots, b_m$ .
- Regras de “simpagação”:  $h_1, h_2, \dots, h_n \setminus h_{n+1}, h_{n+2}, \dots, h_p \Leftrightarrow \text{Guarda} \mid b_1, b_2, \dots, b_m$ .

Cada  $h_i$  e cada  $b_i$  é um predicado especialmente denominado predicado de restrição ou, simplesmente, restrição ( $n \geq 1, m \geq 1$  e  $p \geq 2$ ). Os predicados de restrição  $h_i$  e  $b_i$  formam o que, de maneira correlata às regras de Prolog, chamam-se, respectivamente, de cabeça e corpo da regra. A “;” (virgula) é o operador de sequenciamento e significa conjunção. No corpo de uma regra é possível usar o operador “;” no lugar do operador “,” para significar disjunção entre predicados. O *Guarda* é um conjunto, possivelmente vazio, de predicados separados pelo operador “;”. Um *Guarda* vazio é interpretado como verdade (*true*) e pode deixar de ser descrito.

As regras de um programa CHR podem ser entendidas como regras de reescrita sobre estados, sendo um estado definido pelo conjunto de restrições presentes em um certo instante em um banco de restrições, a partir daqui chamado apenas de banco. Inicialmente o banco é preenchido com a questão a ser avaliada. Uma regra é aplicada se (1) a cabeça da regra coincide com a cabeça de restrições presentes no banco e (2) o guarda da regra é satisfeito. Regras de simplificação implementam uma bi-implicação, trocando um conjunto de restrições no banco por outro conjunto de restrições equivalente. Regras de propagação implementam uma implicação por adicionar novas restrições sem remover seus “antecedentes”. Regras de “simpagação” são uma mistura das outras duas, sendo que as restrições na cabeça, que aparecem antes da barra invertida, permanecem no banco e aquelas que estão após a barra invertida são removidas. A unificação de átomos ocorre como em Prolog. Uma descrição formal da semântica operacional de CHR é apresentada em Duck *et al.* (2004).

**Exemplo 6.1 (Um programa em CHR).** O programa CHR ilustrado na Figura 6.1 estabelece regras para a igualdade de duas variáveis. A linha 0 informa ao compilador Prolog que o programa usará a biblioteca “chr”, disponível no ambiente SWI-Prolog. A

linha 1 declara que o programa usará o predicado de restrição *menorIgual*, que possui aridade 2. As linhas 2, 3, 4 e 5 definem regras que descrevem, respectivamente, as propriedades reflexiva, antissimétrica, idempotente e transitiva da igualdade entre duas variáveis. Todas as regras possuem cabeça, corpo e guarda vazio, o que significa que o guarda é sempre true. As linhas 2 e 3 são exemplos de regras de simplificação. Ou seja, a ocorrência de *menorIgual(X,X)* no banco de restrições deve ser substituída por *true* e, a ocorrência simultânea de *menorIgual(X,Y)* e *menorIgual(Y,X)* deve ser substituída por  $X=Y$ . A linha 4 é um exemplo de uma regra de “simpagação” e descreve que, na ocorrência de duas restrições *menorIgual(X,Y)*, uma delas será removida do banco de restrições. A linha 5 é um exemplo de uma regra de propagação e descreve que, na ocorrência das restrições *menorIgual(X,Y)* e *menorIgual(Y,Z)*, elas devem ser mantidas e uma nova restrição, *menorIgual(X,Z)*, deve ser escrita no banco de restrições.

```

0. :- use_module(library(chr)).
1. :- chr_constraint menorIgual/2.
2. menorIgual(X,X) <=> true. % Regra para a propriedade reflexiva.
3. menorIgual(X,Y) , menorIgual(Y,X) <=> X = Y. % Regra para a propriedade antissimétrica.
4. menorIgual(X,Y) \ menorIgual(X,Y) <=> true. % Regra para a propriedade idempotente.
5. menorIgual(X,Y) , menorIgual(Y,Z) ==> menorIgual(X,Z). % Regra para a propriedade transitiva.

```

**Figura 6.1. Exemplo de um programa CHR** — adaptado de SWI-Prolog-Org (2017). ■

### 6.3 – Semântica operacional de um programa CHR

Operacionalmente um programa CHR usa regras para, passo a passo, modificar um banco, inicialmente em um estado  $B^1$  preenchido com uma questão  $Q$ . Em cada passo, uma regra é escolhida para ser aplicada. A regra a ser escolhida é a primeira da ordem em que são escritas a casar suas cabeças com restrições em  $B$ . A aplicação de uma regra é chamada de derivação. Quando a quantidade de derivações é finita e termina com  $B$  em um estado  $B^n$  ( $n \geq 1$ ), se  $B^n$  for uma contradição então diz-se que as derivações falharam, caso contrário, as derivações foram bem sucedidas e  $B^n$  é o conjunto de restrições que respondem à questão  $Q$ .

### Exemplo 6.2 (Descrição operacional de uma execução do programa da Figura 6.1).

A Figura 6.2 mostra a execução do programa da Figura 6.1 para uma questão  $Q = \text{menorIguar}(X,Y), \text{menorIguar}(Y,Z), \text{menorIguar}(Z,X)$ , descrita na linha 0. O sistema executa  $Q$  e dá como um resultado  $X = Y, X = Z$  (linha 1). O “;” digitado ao final da linha 1 solicita ao sistema que apresente outros resultados. Como neste caso não há, o sistema responde com *false* (linha 2). Operacionalmente, o sistema começa inicializando um banco com as restrições da questão, assim em seu estado 1 é  $B^1 = \{ \text{menorIguar}(X,Y), \text{menorIguar}(Y,Z), \text{menorIguar}(Z,X) \}$ . A regra de transitividade (linha 5, Figura 6.1) é aplicada à primeira e segunda restrição, produzindo a inserção de  $\text{menorIguar}(X,Z)$  a  $B^1$ , que no estado 2 passa a ser  $B^2 = \{ \text{menorIguar}(X,Y), \text{menorIguar}(Y,Z), \text{menorIguar}(Z,X), \text{menorIguar}(X,Z) \}$ . No próximo passo, a regra de antissimetria (linha 3, Figura 6.1) é aplicada à terceira e à quarta restrição de  $B^2$ , para produzir a ligação (*binding*) entre as variáveis  $X$  e  $Z$ , derivando  $B^3 = \{ \text{menorIguar}(X,Y), \text{menorIguar}(Y,Z), X=Z \}$ . Como  $X=Z$  então  $B^3 = \{ \text{menorIguar}(X,Y), \text{menorIguar}(Y,X), X=Z \}$  também. Em seguida, novamente a regra de antissimetria é aplicada à primeira e segunda restrições de  $B^3$ , produzindo a ligação  $X=Y$ , para derivar  $B^4 = \{ X=Y, X=Z \}$ . Como nenhuma regra pode ser aplicada a  $B^4$ , então o conjunto de derivações tem fim e  $\{ X=Y, X=Z \}$  é o conjunto de restrições que responde à questão  $Q$ .

<ol style="list-style-type: none"><li>0. ?- menorIguar(X,Y), menorIguar(Y,Z), menorIguar(Z,X).</li><li>1. X = Y, X = Z ;</li><li>2. <b>false</b>.</li></ol>
---

Figura 6.2. Exemplo de execução do programa da Figura 6.1.

■

## 6.4 – Uso da linguagem CHR para implementar raciocínio abdutivo

A Definição 3.1 estabelece que um raciocínio abdutivo tem como resultado um conjunto  $H$  de hipóteses, calculado sobre um sistema  $\langle T, F \rangle$ , onde  $T$  é o conjunto teoria e  $F$  é o conjunto de fatos. Raciocínios abduativos podem ser implementados em linguagem CHR, no ambiente SWI-Prolog, declarando o conjunto:

- Teoria  $T$  por meio de um programa Prolog que usa as regras de simplificação, propagação e simpagação da linguagem CHR.
- De fatos  $F = \{ f_1, f_2, \dots, f_q \}$ , como uma questão “?-  $f_1, f_2, \dots, f_q$ ” a ser executada pelo ambiente SWI-Prolog.

O resultado da execução de uma questão é o conjunto de hipóteses  $H = \{ h_1, h_2, \dots, h_p \}$ , reportadas pelo ambiente SWI-Prolog na forma:

```

h1
true ;

h2
true ;

...
true ;

hp
true ;

false.

```

De forma detalhada, o programa que descreve uma teoria  $T = \{ t_1, t_2, t_3, \dots, t_m \}$  tem, no ambiente SWI-Prolog, o formato geral ilustrado na Figura 6.3. A linha 0 descreve ao compilador Prolog do ambiente SWI-Prolog que a biblioteca “chr” será utilizada. Na linha 1 devem ser declaradas as restrições que, no presente caso, por se tratar de programa para computar raciocínio abdutivo, são os predicados usados no programa, conhecidos, neste contexto, também como predicados abduíveis. As demais linhas do programa descrevem uma teoria  $T$  via regras de simplificação, propagação e simpagação da linguagem CHR.

```

0. :- use_module(library(chr)).
1. :- chr_constraint ← Lista de restrições (predicados abduíveis).

2.
3. ← Regras de simplificação, propagação e simpagação para
4. descrever t1, t2, t3, ..., tm.
...

```

**Figura 6.3. Formato geral de um programa CHR para implementar raciocínio abdutivo no ambiente SWI-Prolog.**

Programas para abdução podem também misturar sentenças CHR com sentenças Prolog, possibilitando, com isto, acesso às facilidades expressivas de Prolog. Embora isto possa ser feito, este trabalho tem foco, apenas, nas sentenças em linguagem CHR.

O Exemplo 6.3, apresentado a seguir, mostra um programa CHR para computar raciocínio abdutivo. Especificamente, o programa modela uma teoria sobre um sistema de partida de motor à combustão de automóvel. O exemplo contempla, também, a execução de raciocínios sobre possíveis defeitos ou funcionalidades sobre o sistema de partida de motor à combustão de automóvel.

**Exemplo 6.3 (Um programa CHR para computar raciocínio abdutivo sobre um sistema de partida de motor à combustão de automóvel).** Sistemas para dar partida no motor à combustão de automóveis são, hoje em dia, muito sofisticados e possuem muitas diferenças entre marcas de automóveis, e este exemplo não tem a intenção de capturar toda a sofisticação e variedade existentes. O exemplo se limita ao funcionamento e aos defeitos de alguns poucos, todavia importantes, componentes usualmente presentes na maioria dos sistemas de partida de motor à combustão de automóvel. Uma teoria  $T$  para modelar um sistema de partida pode ser assim descrita:

$$\begin{aligned}
 T = \{ & \\
 & \textit{bateriaSemCarga} \vee \textit{motorArranqueComDefeito} \vee \textit{comFalhasAlimentacao} \rightarrow \\
 & \textit{motorNaoDaPartida} , \\
 & \textit{bateriaSemCarga} \vee \textit{motorArranqueComDefeito} \rightarrow \textit{motorNaoDaPartida} , \\
 & \textit{bateriaSemCarga} \rightarrow \textit{luzesPainelApagadas} , \\
 & \textit{bateriaComCarga} \wedge \textit{motorArranqueSemDefeito} \wedge \textit{semFalhasAlimentacao} \rightarrow \\
 & \textit{motorDaPartida} , \\
 & \textit{bateriaComCarga} \wedge \textit{motorArranqueSemDefeito} \rightarrow \textit{motorArranqueFazRuido} , \\
 & \textit{bateriaComCarga} \rightarrow \textit{luzesPainelAcesas} , \\
 & \textit{bateriaSemCarga} \rightarrow \neg \textit{bateriaComCarga} , \\
 & \textit{motorArranqueComDefeito} \rightarrow \neg \textit{motorArranqueSemDefeito} , \\
 & \textit{comFalhasAlimentacao} \rightarrow \neg \textit{semFalhasAlimentacao} , \\
 & \textit{motorNaoDaPartida} \rightarrow \neg \textit{motorDaPartida} , \\
 & \textit{luzesPainelApagadas} \rightarrow \neg \textit{luzesPainelAcesas} , \\
 & \textit{motorArranqueNaoFazRuido} \rightarrow \neg \textit{motorArranqueFazRuido} \\
 & \}.
 \end{aligned}$$

A teoria está descrita por meio de predicados com aridade igual a zero, que nomeiam proposições mnemônicas para “motor dar ou não partida”, “bateria com e sem carga”, “motor de arranque com e sem defeito”, “sistema de alimentação com e sem falha”, “motor de arranque demonstrando bom funcionamento por meio de ruído característico ao ser acionado, ou não” e “luzes do painel acesas ou apagadas”. As três primeiras sentenças de  $T$  definem relações lógicas para os defeitos ou mau funcionalidades que levam ao motor não dar partida e, as três sentenças seguintes, definem relações lógicas para as funcionalidades que levam ao motor dar partida. As seis últimas sentenças tratam apenas de definir relações lógicas antônimas entre predicados: *bateriaSemCarga* é o antônimo de *bateriaComCarga*, *motorArranqueComDefeito* é o antônimo de *motorArranqueSemDefeito* etc..

```

0. :- use_module(library(chr)).
1. :- chr_constraint bateriaSemCarga, bateriaComCarga,
   motorArranqueComDefeito, motorArranqueSemDefeito,
   comFalhasAlimentacao, semFalhasAlimentacao,
   motorNaoDaPartida, motorDaPartida,
   luzesPainelApagadas, luzesPainelAcesas,
   motorArranqueFazRuido, motorArranqueNaoFazRuido.

2. bateriaSemCarga \ bateriaSemCarga <=> true. bateriaComCarga \ bateriaComCarga <=> true.
3. motorArranqueComDefeito \ motorArranqueComDefeito <=> true.
4. motorArranqueSemDefeito \ motorArranqueSemDefeito <=> true.
5. comFalhasAlimentacao \ comFalhasAlimentacao <=> true.
6. semFalhasAlimentacao \ semFalhasAlimentacao <=> true.
7. motorNaoDaPartida \ motorNaoDaPartida <=> true. motorDaPartida \ motorDaPartida <=> true.
8. luzesPainelApagadas \ luzesPainelApagadas <=> true. luzesPainelAcesas \ luzesPainelAcesas <=> true.
9. motorArranqueFazRuido \ motorArranqueFazRuido <=> true.
10. motorArranqueNaoFazRuido \ motorArranqueNaoFazRuido <=> true.

11. bateriaSemCarga , bateriaComCarga <=> fail.
12. motorArranqueComDefeito, motorArranqueSemDefeito <=> fail.
13. comFalhasAlimentacao, semFalhasAlimentacao <=> fail.
14. motorNaoDaPartida, motorDaPartida <=> fail.
15. luzesPainelApagadas, luzesPainelAcesas <=> fail.
16. motorArranqueNaoFazRuido, motorArranqueFazRuido <=> fail.

17. motorNaoDaPartida <=> bateriaSemCarga ; motorArranqueComDefeito ; comFalhasAlimentacao.
18. motorArranqueNaoFazRuido <=> bateriaSemCarga ; motorArranqueComDefeito.
19. luzesPainelApagadas <=> bateriaSemCarga.
20. motorDaPartida <=> bateriaComCarga , motorArranqueSemDefeito , semFalhasAlimentacao.
21. motorArranqueFazRuido <=> bateriaComCarga , motorArranqueSemDefeito.
22. luzesPainelAcesas <=> bateriaComCarga.

```

**Figura 6.4. Programa em linguagem CHR modelando uma teoria sobre um sistema de partida de motor à combustão de automóvel.**

A Figura 6.4 apresenta um programa, em linguagem CHR, implementando a teoria  $T$ . A linha 0 especifica que a biblioteca “chr” será utilizada e a linha 1 declara cada um dos predicados, empregados em  $T$ , como um predicado abduzível. As seis primeiras sentenças de  $T$  estão “traduzidas” nas linhas 17 a 22 da Figura 6.4; as seis últimas sentenças de  $T$  estão “traduzidas” nas linhas 11 a 16 da Figura 6.4. O conjunto de sentenças, apresentadas nas linhas 2 a 10, têm por finalidade a eliminação de duplicatas de um mesmo predicado, nas hipóteses que serão calculadas pelo programa. A ocorrência de duplicatas é um efeito comum em CHR, devido à maneira como a semântica operacional da linguagem é definida, todavia nem sempre é desejável, como neste caso. Assim, regras de simplificação como a primeira regra da linha 2, especifica que na ocorrência de dois predicados *bateriaSemCarga*, o primeiro permanece no banco de restrições enquanto o segundo é removido.

A Figura 6.5 apresenta três raciocínios abduzitivos, realizados com o programa da Figura 6.4, com o objetivo de produzir hipóteses para o fato do motor à combustão não dar partida, considerando, em cada caso, fatos adicionais.

```

0. ?- motorNaoDaPartida , luzesPainelAcesas.
1.  bateriaComCarga
2.  motorArranqueComDefeito
3.  true ;
4.  bateriaComCarga
5.  comFalhasAlimentacao
6.  true ;
7.  false.

8. ?- motorNaoDaPartida , motorArranqueFazRuido.
9.  bateriaComCarga
10. motorArranqueSemDefeito
11. comFalhasAlimentacao
12. true ;
13. false.

14. ?- motorNaoDaPartida , motorArranqueNaoFazRuido , luzesPainelAcesas.
15. bateriaComCarga
16. motorArranqueComDefeito
17. true ;
18. bateriaComCarga
19. motorArranqueComDefeito
20. comFalhasAlimentacao
21. true ;
22. false.

```

**Figura 6.5. Execução de três raciocínios abduzitivos com o programa da Figura 6.4.**

O primeiro raciocínio abduativo, expresso nas linhas de 0 a 7, apresenta como fatos o motor não dar partida (*motorNaoDaPartida*) e as luzes do painel estarem acesas (*luzesPainelAcesas*). Este raciocínio produz, como resposta, o seguinte conjunto contendo duas hipóteses:

$$\{ \textit{bateriaComCarga} \wedge \textit{motorArranqueComDefeito} , \\ \textit{bateriaComCarga} \wedge \textit{comFalhasAlimentacao} \}.$$

O segundo raciocínio abduativo, expresso nas linhas de 8 a 13, apresenta juntamente com fato do motor não dar partida, o fato do motor de arranque fazer o ruído típico de seu acionamento (*motorArranqueFazRuido*). Apenas uma hipótese é calculada neste caso, conduzindo, como resposta, ao conjunto unitário,

$$\{ \textit{bateriaComCarga} \wedge \textit{motorArranqueSemDefeito} \wedge \textit{comFalhasAlimentacao} \}.$$

O terceiro raciocínio abduativo, expresso nas linhas 14 a 22, apresenta dois fatos adicionalmente ao fato do motor não dar partida: o fato do motor de arranque não fazer ruído (*motorArranqueNaoFazRuido*) e o fato das luzes do painel estarem acesas (*luzesPainelAcesas*). O resultado deste raciocínio é o conjunto de hipóteses

$$\{ \textit{bateriaComCarga} \wedge \textit{motorArranqueComDefeito} , \\ \textit{bateriaComCarga} \wedge \textit{motorArranqueComDefeito} \wedge \textit{comFalhasAlimentacao} \}.$$

■

# Capítulo 7

## Pesquisa Experimental sobre Atividade de Modelagem Computacional Educacional Abdutiva

*Este Capítulo descreve o planejamento e a realização de uma pesquisa experimental, em laboratório, de uma atividade de modelagem computacional educacional abdutiva. O objetivo desta pesquisa é identificar facilidades e dificuldades na realização da atividade de modelagem computacional educacional abdutiva, instanciada da maneira proposta neste trabalho. A Seção 7.1 apresenta uma visão geral sobre a metodologia da pesquisa. Em seguida, são descritos os materiais empregados (Seção 7.2), os procedimentos experimentais (Seção 7.3), os sujeitos da pesquisa (Seção 7.4) e os critérios para análise quantitativa e qualitativa dos resultados (Seção 7.5). Os resultados são apresentados e discutidos na Seção 7.6.*

## 7.1 – Visão geral da metodologia da pesquisa

Esta é uma pesquisa experimental, realizada em laboratório, com o cuidado e protocolo habitual que se presta ao planejamento e execução de uma pesquisa experimental, em conformidade com o que, a exemplo, é descrito em Lazar, Feng & Hochheiser (2010). Esta pesquisa é planejada com as devidas especificidades relativas à atividade de modelagem computacional educacional abdutiva, como é descrito a seguir.

Destaca-se no ambiente laboratorial da pesquisa:

- O pesquisador.
- Os sujeitos da pesquisa.
- As diretrizes para modelagem computacional abdutiva (Definição 5.1, Capítulo 5).
- As estações de trabalho.
- O conjunto de atividades a serem realizadas.

Sendo esta uma pesquisa sobre modelagem computacional com propósito educacional, é necessário situar os papéis do pesquisador e dos sujeitos da pesquisa, posto que, neste laboratório, que simula um ambiente para modelagem computacional, os sujeitos exercem também o papel de estudantes (aprendizes) e o pesquisador também exerce o papel de mediador, atuando em conformidade com o que é esperado de um mediador em um ambiente para modelagem (cf. Capítulo 2). Ou seja, o pesquisador não se limita a fazer observações e anotações sobre o que está acontecendo nas sessões experimentais, mas atua mediando as relações dos sujeitos da pesquisa — que exercem o papel de estudantes — com atividades de modelagem. O pesquisador–mediador, não soluciona a atividade para os sujeitos–estudantes, mas os motiva e indica-lhes caminhos para as soluções. Isto é planejado desta forma, porque se quer que o ambiente laboratorial da pesquisa reflita um ambiente escolar planejado, pedagogicamente, para atividades de modelagem.

A Definição 5.1, que declara as diretrizes para modelagem computacional educacional abdutiva, descreve em detalhes a atividade que deve ser realizada no laboratório. Esta atividade é expressa por um variado conjunto de ações que demanda do sujeito–

estudante da pesquisa, diversas habilidades para que modelos sejam desenvolvidos. Tendo por fundamento a Definição 5.1, o sujeito–estudante deve, para realizar atividade de modelagem, exercitar as seguintes habilidades:

- H0: Habilidade para **motivar-se**, inspirar-se, manifestar a vontade de criar um modelo para algo — um fenômeno, uma máquina, um objeto — e definir qual é a sua intenção de modelagem — estabelecer um modelo realístico, surrealístico etc. (ação anterior ao processo de modelagem, é implícita à Definição 5.1).
- H1: Habilidade para **investigar** algo quer se quer modelar, i.e., realizar estudos, buscar informações sobre o que se quer modelar (ação anterior ao processo de modelagem, é implícita à Definição 5.1).
- H2: Habilidade para **abstrair** conceitos sobre algo que se quer modelar, i.e., selecionar elementos — características, propriedades, proposições, normas, regras etc. — que farão parte do modelo a ser estabelecido (parte da ação 1 da Definição 5.1).
- H3: Habilidade para **descrever** o modelo computacionalmente, i.e., habilidade para escrever o modelo usando uma linguagem que, no caso deste experimento, é a linguagem CHR (parte da ação 1 da Definição 5.1).
  - H3.1: Habilidade para criar predicados para representar os conceitos selecionados que irão fazer parte do modelo.
  - H3.2: Habilidade para relacionar os predicados por meio de regras da linguagem CHR.
- H4: Habilidade para **propor fatos e executar** em computador o modelo desenvolvido (ações 2 e 3 da Definição 5.1).
- H5: Habilidade para **avaliar** os resultados da execução de um modelo (ações 4 e 5 da Definição 5.1).
  - H5.1: Habilidade para **ler e interpretar** os resultados da execução de um modelo.
  - H5.2: Habilidade para **julgar** o resultado da execução de um modelo em relação a uma intenção de modelagem.

Pretende-se com esta pesquisa investigar facilidades e dificuldades na realização da atividade de modelagem computacional educacional abduativa, instanciada da maneira como se descreverá a seguir. Tais facilidades e dificuldades foram inferidas quantitativamente e qualitativamente, indiretamente pela resposta do sujeito–estudante a desafios que solicitam o exercício de cada uma das habilidades necessárias à modelagem computacional educacional abduativa, descritas anteriormente nesta seção.

As habilidades H0 e H1 não integram esta pesquisa, pois, o pesquisador–mediador determinou o que foi modelado e deu informações sobre o funcionamento do que foi modelado. Excetuando-se estas duas habilidades, a investigação de todas as demais habilidades integram a pesquisa.

Cada uma das estações de trabalho é equipada com mesa, cadeira, computador e software adequado ao experimento; mais à diante a configuração da estação de trabalho será detalhada.

O conjunto de atividades a serem realizadas é planejado para desafiar o sujeito–estudante ao exercício das habilidades H2, H3, H4 e H5, e serão discutidas, em detalhes na próxima seção.

A pesquisa está dividida em cinco fases que, sinteticamente, objetivam:

- Fase 1: O delineamento do perfil dos participantes.
- Fase 2: A resolução de exercícios que requerem habilidade para abstrair conceitos (habilidade H2).
- Fase 3: A escrita de um modelo usando a linguagem CHR (habilidade H3).
- Fase 4: Exercitar a habilidade de propor fatos para execução computacional de modelos (habilidade H4).
- Fase 5: Exercitar a habilidade de avaliar os resultados da execução de um modelo (habilidade H5).

## **7.2 – Materiais**

Desenvolveu-se um formulário para assistir a todas as fases das sessões experimentais da pesquisa. O formulário foi validado previamente em um experimento piloto que contou

com a participação de duas pessoas, sendo que pequenos problemas foram removidos e o formulário foi melhorado. O Apêndice I descreve integralmente o Formulário e esta seção discute o seu conteúdo.

O Formulário é constituído por cinco partes, cada uma delas projetada para assistir a uma fase específica do procedimento experimental, que será descrito na próxima seção. O formulário não solicita a identificação pessoal do participante<sup>8</sup>; apenas um código numérico é atribuído às partes do formulário para que se possa agrupar correspondentemente as respostas dadas pelos participantes.

A Parte 1 explica ao participante o propósito do experimento e permite que ele declare o seu perfil. O perfil (itens 1 a 4 do Formulário) está sendo caracterizado pelo sexo, idade, curso e turma do participante. O curso e a turma do participante integram o perfil por que, como será descrito posteriormente, a pesquisa foi realizada em uma escola.

A Parte 2 contém exercícios que requerem do participante a habilidade para abstrair conceitos. No início da Parte 2 são apresentadas três fotografias de um Jeep: a primeira retrata o Jeep por um ângulo dianteiro e lateral; a segunda retrata o painel de instrumentos, com destaques textuais para o botão que permite ligar e desligar os faróis e para o interruptor da ignição; a terceira retrata o compartimento do motor, com destaques textuais para a bateria e para o motor de arranque. Como se verá, posteriormente, a apresentação destas fotografias foi complementada por informações verbais, dadas pelo pesquisador, sobre o funcionamento de alguns sistemas do Jeep. Os exercícios desta parte do formulário solicitam que o participante descreva termos e proponha relações lógicas entre os termos, considerando que a intenção é estabelecer um modelo que explique por que:

- O farol esquerdo e/ou o farol direito do Jeep estão apagados ou acesos (itens de 5 a 26 do Formulário).
- O motor do Jeep dá ou não dá partida (itens de 27 a 42 do Formulário).

A Parte 3 contém um exercício (item 43 do Formulário) que requer do participante a habilidade de desenvolver, usando a linguagem CHR, um modelo que explique por que

---

<sup>8</sup> O termo participante está sendo usado também para se referir ao sujeito da pesquisa.

o farol esquerdo e/ou o farol direito do Jeep estão apagados ou acesos, a partir de termos e relações lógicas entre termos, indicados no exercício. A ideia é que o participante escreva (programe) em linguagem CHR, interprete, execute e modifique o modelo tantas vezes quanto quiser para, ao final, entregar como resultado um arquivo contendo o modelo. Para facilitar o trabalho do participante, ele recebe o arquivo “MeuModelo.pl”, gravado na área de trabalho, já com o cabeçalho do modelo previamente preenchido (seções “:- use\_module(library(chr)” e “:- chr\_constraint”). Como se verá mais adiante, a resolução do exercício da Parte 3 foi precedida por uma explicação, pelo pesquisador, sobre a linguagem CHR e sobre o ambiente que a implementa, o SWI-Prolog.

A Parte 4 contém exercícios (itens 44, 46, 48 e 50 do Formulário) que requerem do participante a habilidade para propor fatos para execução de modelos. Ele recebe, já pronto, um arquivo contendo um modelo que explica o fato dos faróis do Jeep estarem apagados ou acesos.

A Parte 5 do Formulário contém exercícios que requerem do participante a habilidade para avaliar os resultados da execução de um modelo, i.e., habilidade para ler, interpretar e julgar os resultados da execução de um modelo em relação à uma intenção de modelagem. Duas séries de exercícios são propostos nesta parte. A primeira série de exercícios (itens 45, 47, 49 e 51) solicita a execução para os fatos que ele descreveu nos exercícios 44, 46, 48 e 50, em computador, de um modelo, que o participante recebe, já pronto. Esta série de exercícios foi cuidadosamente planejada para que surja situações em que a execução do modelo:

- Gera hipóteses que explicam os fatos apresentados.
- Não gera hipóteses para os fatos apresentados.
- Gera, apenas, hipóteses não explanatórias, i.e., o conjunto resultante de hipóteses é igual ao conjunto de fatos apresentados.

Na segunda série de exercícios (itens 52 a 56 do Formulário) são apresentadas solicitações de execução de modelos com respectivos resultados da execução; em cada exercício, solicita-se do participante o assinalamento de uma alternativa que ele considera correta, entre várias disponíveis.

### 7.3 – Procedimento experimental

A pesquisa é executada em sessões experimentais com a participação de, no máximo, cinco sujeitos, em cinco estações de trabalho de um laboratório, sendo que cada estação é equipada com uma mesa, uma cadeira, um computador com processador Intel i3, 3.3 GHz, 4 GB de memória RAM, executando Windows 10 Pro, um editor de textos e um compilador para a linguagem CHR, disponível por meio do ambiente SWI-Prolog.

Cada sessão experimental é composta por seis fases, detalhadas a seguir.

A Fase 1, programada para ser realizada em 5 minutos, inicia com o acolhimento, cumprimento e agradecimento aos participantes pela contribuição voluntária e gratuita. Com o auxílio de um projetor multimídia, o pesquisador expõe o propósito do experimento, distribui a Parte 1 do Formulário e solicita aos participantes que respondam às questões relacionadas ao seu perfil. Após os participantes responderem às questões da Parte 1 do Formulário, o pesquisador recolhe as respostas.

A Fase 2, exercícios que requerem do participante a habilidade para abstrair conceitos, programada para ser realizada em 20 minutos, inicia com a distribuição da Parte 2 do Formulário. Em seguida, de forma verbal e com o auxílio de um projetor multimídia, o pesquisador apresenta o Jeep, ilustrado nas fotos da Parte 2, como o artefato que foi objeto do exercício sobre abstração. O pesquisador dá explicações sobre o funcionamento do sistema de faróis e também sobre o sistema arranque do Jeep, e solicita que os participantes respondam aos exercícios apresentados no Formulário. Após os participantes responderem aos exercícios da Parte 2 do Formulário, o pesquisador recolhe as respostas.

A Fase 3, exercício que requer habilidade para escrever um modelo usando a linguagem CHR, programada para ser realizada em 60 minutos, inicia com a distribuição da Parte 3 do Formulário aos participantes, e com uma apresentação sobre a linguagem CHR. Com o auxílio de um projetor multimídia, o pesquisador explica aos participantes como escrever cada um dos três tipos de regras da linguagem CHR e utiliza, como exemplo, um modelo que explica por que o motor do Jeep dá ou não dá partida. São dadas também explicações sobre como interpretar e executar um modelo, escrito em linguagem CHR,

no ambiente SWI-Prolog. A estratégia pedagógica é a da aprendizagem através de exemplos, os quais são abordados sem formalizações teóricas. Após isto, o pesquisador solicita que os participantes criem o modelo solicitado no exercício e que façam isto usando as relações apresentadas. Durante o desenvolvimento do modelo, o pesquisador atua como um mediador de um ambiente de modelagem, orientando, discutindo problemas, sem contudo apresentar soluções. Os participantes devem exercitar o ciclo de edição ou reedição do modelo, execução e avaliação de resultados, até que deem o modelo por terminado, ou até o término do tempo da fase, anunciado pelo pesquisador. Ao final da fase, o pesquisador recolhe, em um *pen drive*, os arquivos contendo os modelos desenvolvidos pelos participantes.

A Fase 4, exercícios que requerem habilidade para propor fatos para execução de modelos, programada para ser realizada em 10 minutos, inicia com a distribuição da Parte 4 do Formulário aos participantes. O pesquisador entrega a cada participante um arquivo contendo um modelo que computa explicações para os faróis do Jeep estarem apagados ou acesos, e solicita que eles façam o que é pedido no Formulário. Após responderem aos exercícios da fase, o pesquisador recolhe as respostas à Parte 4 do formulário.

A Fase 5, exercícios sobre a habilidade de avaliar os resultados da execução de um modelo, programada para ser realizada em 15 minutos, inicia com a distribuição da Parte 5 do Formulário. A primeira série de exercícios desta parte é realizada executando-se, em computador, o modelo que computa explicações para os faróis do Jeep estarem apagados ou acesos. A segunda série de exercícios requer, para cada exercício, o assinalamento de uma alternativa entre as múltiplas alternativas disponíveis. Após a conclusão dos exercícios, o pesquisador recolhe as respostas à Parte 5 do Formulário.

A sessão experimental termina com um *debriefing*, com duração de 5 minutos, no qual o pesquisador faz comentários gerais sobre a solução dos exercícios, possibilita que, opcionalmente, os participantes comentem suas soluções, e finaliza a sessão com um agradecimento pelas contribuições.

## 7.4 – Sujeitos

Um total de 88 participantes (Tabela 7.1) de curso técnico integrado ao ensino médio, oferecido pelo Instituto Federal de Educação, Ciência e Tecnologia do Sul de Minas Gerais, Campus Muzambinho, atuam no experimento, num período total de 45 dias. A população de interesse desta pesquisa é constituída por 160 estudantes. O tamanho mínimo da amostra, considerando que foram empregados na análise os métodos estatísticos de análise de variância ANOVA e testes *t* de Student (McClave & Sincich, 2014), é de 75 participantes. Este cálculo foi realizado considerando os seguintes parâmetros: tamanho da população igual a 160, nível de significância igual a 5%, nível de confiança igual a 95% e heterogeneidade-homogeneidade da população igual a 90%. Os participantes são jovens que buscam uma formação técnica simultaneamente a uma formação no ensino médio clássico, aproximadamente metade é do sexo feminino e metade é do sexo masculino, têm idade variando entre 16 e 19 anos, média e mediana de idades igual a 17 anos.

**Tabela 7.1. Gênero e idade dos participantes.**

Gênero, <i>n</i> (%)		Idade, em anos		
Feminino	Masculino	Intervalo	Média	Mediana
41 (47)	47 (53)	16 - 19	17	17

## 7.5 – Critérios para análise quantitativa e qualitativa dos resultados da pesquisa experimental

Os resultados da pesquisa foram analisados de forma:

- Quantitativa: a partir do conjunto de critérios quantitativos que serão descritos a seguir.
- Qualitativa: a partir (1) das anotações realizadas pelo pesquisador, nas diversas sessões experimentais, (2) na leitura e interpretação dos resultados apresentados, dando ênfase para as normalidades, e buscando, também, delinear os extremos das respostas aos exercícios propostos.

O Apêndice II apresenta o padrão de resposta, “gabarito”, para os exercícios propostos nas sessões da pesquisa experimental. Este padrão foi utilizado na análise quantitativa como um sinalizador para o que se considera ser correto. A Tabela 7.2 descreve os critérios, métrica e escala de valores para análise quantitativa dos resultados da pesquisa experimental. As respostas ao(s) exercício(s) proposto(s) na:

- Fase 2, exercícios que requerem habilidade para abstrair conceitos, foram analisados segundo à completude dos conceitos e sentenças abstraídas, consideradas pertinentes, em relação ao padrão de respostas do Apêndice II.
- Fase 3, exercício que requer habilidade para descrever um modelo usando a linguagem CHR, foi analisado segundo a corretude sintática e semântica do modelo digital desenvolvido, executável em computador.
- Fase 4, exercícios que requerem a habilidade para propor fatos e executar um modelo, foram analisados segundo a corretude dos fatos propostos em relação ao padrão de respostas do Apêndice II.
- Fase 5, exercícios que requerem habilidade para avaliar os resultados da execução de um modelo, foram avaliados segundo a corretude da resposta em relação ao padrão de respostas do Apêndice II.

**Tabela 7.2. Critérios, métrica e escala de valores para análise quantitativa dos resultados da pesquisa experimental.**

<b>Critério de análise (Escala)</b>	<b>Métrica</b>
<i>Exercícios da Parte 2 – Habilidade para abstrair conceitos</i>	
Completude dos conceitos abstraídos e das sentenças relacionando os conceitos abstraídos (numérica, intervalo [0..10])	<p>Julga-se se os conceitos abstraídos pertencem ao sistema que se solicita modelar (sistema de faróis ou sistema de partida de motor à combustão do Jeep). Considera-se completo o padrão de resposta descrito no Apêndice II, Parte 2, além de outros julgados pertinentes. Seja <math>\alpha</math> a quantidade de conceitos abstraídos que sejam pertinentes (ou quantidade de sentenças usando os conceitos abstraídos que sejam pertinentes) e seja <math>\beta</math> a respectiva quantidade de conceitos (ou sentenças) escrita no padrão. Atribui-se:</p> <p>0: quando <math>\alpha &lt; 0.5 \beta</math> (incompleto).            1: quando <math>\alpha \geq 0.5 \beta</math> (completo).</p> <p>Somam-se os valores atribuídos aos exercícios e multiplica-se esta soma por 10/8 para obter a nota dos exercícios normalizada para a escala no intervalo [0..10].</p>
<i>Exercício da Parte 3 – Habilidade para descrever um modelo usando a linguagem CHR</i>	
Corretude sintática e semântica do modelo desenvolvido (numérica, intervalo [0..10])	Compila-se o modelo e verifica-se a existência ou não de erros de sintaxe. Em caso de haver erro de sintaxe, atribui-se o valor 0 (zero). Não havendo erro de sintaxe, compara-se o resultado da

execução do modelo com o resultado da execução do modelo padrão, descrito no Apêndice II, Parte 3, para cada um dos dez seguintes raciocínios<sup>9</sup> descritos pelos fatos:

1. *farolEsquerdoAceso*.
2. *farolEsquerdoApagado*.
3. *farolDireitoAceso*.
4. *farolDireitoApagado*.
5. *farolEsquerdoAceso* , *farolDireitoAceso*.
6. *farolEsquerdoAceso* , *farolDireitoApagado*.
7. *farolEsquerdoApagado* , *farolDireitoAceso*.
8. *farolEsquerdoApagado* , *farolDireitoApagado*.
9. *bateriaSemCarga*.
10. *farolEsquerdoAceso* , *farolDireitoAceso* , *bateriaSemCarga*.

Soma-se 1 para a cada resultado do modelo que coincidir com o resultado do modelo padrão.

---

***Exercícios da Parte 4 – Habilidade para propor fatos para executar modelos***

Corretude dos fatos propostos e resultado da execução  
(numérica, intervalo [0..10])

Compara-se a resposta apresentada com a resposta assumida como correta no padrão de respostas descrito no Apêndice II, Parte 4. Para cada um dos 4 exercícios propostos (44, 46, 48 e 50), atribui-se o valor 1 quando há coincidência entre as respostas e 0, caso contrário.

Somam-se os valores atribuídos às respostas e multiplica-se esta soma por 10/4 para obter a nota dos exercícios normalizada para a escala no intervalo [0..10].

---

***Exercícios da Parte 5 – Habilidade para avaliar os resultados da execução de modelos***

Corretude da avaliação dos resultados da avaliação de modelos  
(numérica, intervalo [0..10])

Compara-se a resposta dada com a resposta assumida como correta no padrão de respostas descrito no Apêndice II, Parte 5.

Para cada um dos 9 exercícios propostos (45, 47, 49, 51, 52, 53, 54, 55 e 56), atribui-se o valor 1 quando há coincidência entre as respostas e 0, caso contrário.

Somam-se os valores atribuídos aos exercícios e multiplica-se esta soma por 10/9 para obter a nota dos exercícios normalizada para a escala no intervalo [0..10].

---

## 7.6 – Resultados e discussão

A Tabela 7.3 apresenta a média e desvio padrão das notas obtidas pelos participantes em cada uma das fases do experimento.

---

<sup>9</sup> Sabendo-se que o conjunto de predicados abduzíveis do modelo apresentado como padrão de resposta no Apêndice II, Parte 3, tem 12 elementos, então a quantidade total de possíveis conjuntos de fatos a serem formulados para execução é igual a  $2^{12} - 1$  (-1 para retirar o conjunto vazio de fatos) = 4095. A quantidade de conjuntos de fatos aqui selecionados, 10, é bem menor do que 4095, porém é abrangente porque engloba: (1) situações típicas envolvendo raciocínio que levam em consideração signos – faróis esquerdo ou direito acesos ou apagados, bateria com ou sem carga – que seriam comumente usados em um raciocínio típico com o objetivo de diagnosticar falhas ou funcionalidades no sistema de faróis de um automóvel; (2) situações em que a resposta ao raciocínio é um conjunto de hipóteses (2.1) vazio, (2.2) não vazio (2.3) não explanatório, i.e., é igual ao conjunto de fatos.

**Tabela 7.3. Resultados (média e desvio padrão) das notas obtidas pelos participantes.**

Fase	<i>n</i>	Média (0 – 10) ± DV
2	88	5.75 ± 2.57
3	88	8.27 ± 2.66
4	88	9.35 ± 1.56
5	88	7.46 ± 1.90

Estes resultados sugerem um desempenho regular dos participantes na Fase 2, exercícios sobre a habilidade para abstrair conceitos, e um bom desempenho dos participantes nas demais fases do experimento.

Todavia, foi investigada a questão: o desempenho dos participantes em todas as fases é o mesmo, sendo a variação das médias uma casualidade? Para esta investigação realizou-se uma análise estatística, definindo-se as hipóteses:

- $H_0$ : Não existe diferença entre as médias obtidas nas fases do experimento.
- $H_A$ : Existe diferença entre as médias obtidas nas fases do experimento.

Decidiu-se realizar um teste de análise de variância (ANOVA), com tratamento *a posteriori* por testes *t de Student* para dados emparelhados, com nível de significância igual a 0.05 (5%).

Para a população estudada, o Teste ANOVA sugere que existe diferença significativa entre as médias ( $F(3, 348) = 40.82, p < 0.0001$ ), ou seja, deve-se rejeitar a hipótese nula  $H_0$  e aceitar-se a hipótese alternativa  $H_A$ . Testes *t de Student* realizados, sugerem que existe diferença significativa entre as médias, comparadas duas a duas:

- Fase 2 e Fase 3 ( $t = 7.51, p < 0.01$ ).
- Fase 2 e Fase 4 ( $t = 10.71, p < 0.01$ ).
- Fase 2 e Fase 5 ( $t = 5.09, p < 0.01$ ).
- Fase 3 e Fase 4 ( $t = 3.20, p < 0.01$ ).
- Fase 3 e Fase 5 ( $t = 2.42, p < 0.01$ ).

- Fase 4 e Fase 5 ( $t = 5.62, p < 0.01$ ).

Ou seja, embora o desempenho da população estudada tenha sido regular na Fase 2, boas nas fases 3, 4 e 5, existe uma diferença, estatisticamente significativa, entre o desempenho obtido pelos participantes em cada uma das fases. Em ordem decrescente, considerando o valor da média (Tabela 7.3), temos o desempenho nos exercícios que requerem habilidade para:

- (1) Propor fatos para execução de modelos (Fase 4);
- (2) Descrever um modelo usando a linguagem CHR (Fase 3);
- (3) Avaliar os resultados da execução de modelos (Fase 5); e
- (4) Abstrair conceitos (Fase 2).

Entre as habilidades investigadas, não é surpresa constatar que a de melhor desempenho foi a de proposta de fatos (Fase 4), pois requer simplesmente a seleção, lendo um modelo, de termos a serem usados como fatos.

Por outro lado, é surpreendentemente positivo constatar que a população estudada superou bem os desafios da escrita de sentenças na linguagem CHR, habilidade exigida na Fase 3.

O desempenho na fase de avaliação de resultados da execução de modelos (Fase 5) foi menor do que o desempenho nas fases de proposta de fatos e de escrita de sentenças na linguagem CHR. Pode-se conjecturar que a enorme diferença, entre a linguagem natural e a linguagem usada pelo ambiente SWI-Prolog para apresentar resultados, tenha afetado negativamente o desempenho na fase de avaliação de resultados da execução de modelos. Por exemplo, a sentença apresentada pelo sistema SWI-Prolog, mostrada nas linhas de 1 até 7 da Figura 6.5, é muito diferente de algo que, em linguagem natural ficaria parecido com “as hipóteses computadas são: (1) a bateria está com carga e o motor de arranque está com defeito; ou (2) a bateria está com carga e existem falhas no sistema de alimentação”.

Para a população estudada, a habilidade para abstrair conceitos destaca-se entre as habilidades necessárias para modelagem. O destaque, neste caso, refere-se ao significativo menor desempenho apresentado nesta habilidade. Lamentavelmente, esta

habilidade é crítica no processo de modelagem, porque o eventual insucesso na atividade de abstrair conceitos tem forte influência negativa sobre as demais atividades do processo de modelagem. Dificuldades com abstração de conceitos podem prejudicar e, até mesmo, impedir um estudante de desenvolver um modelo, e isto pode causar a desmotivação do estudante com a modelagem. No estudo realizado, a habilidade de abstrair conceitos foi estudada de forma independente das demais, de maneira que o insucesso nesta habilidade não tinha efeito negativo sobre outras atividades do processo de modelagem. Todavia, os resultados sugerem que, para a população estudada, a habilidade de abstrair conceitos é o “ponto crítico” dentro de um cenário de aplicação da modelagem como prática pedagógica escolar.

Qualitativamente, quais são os resultados? Levando-se em conta as interações com os participantes, durante as sessões experimentais, os pesquisadores interpretam que os participantes envolveram-se nas atividades propostas com entusiasmo e demonstraram gratidão pelo que a atividade de modelagem abstrativa lhes proporcionou em termos de aprendizagem. A liberdade para criar modelos, por um lado, levou o participante a demonstrar uma relação de afeto com o modelo que ele criou. Por outro lado, tal liberdade conduziu a diferentes soluções, muitas delas surpreendentemente inéditas para os pesquisadores.

É preciso observar, todavia, que o experimento proposto aborda modelos que, por simplicidade, se limitam à Lógica Proposicional (predicados com aridade igual a zero). São modelos apropriados para serem apresentados a iniciantes, sem experiência em modelagem usando a linguagem CHR, como é o caso de todos os participantes da pesquisa experimental. Modelos mais complexos, envolvendo, por exemplo, predicados com aridade maior do que zero ou circularidade lógica entre predicados, não é tratado na pesquisa. Isto não desvaloriza esta pesquisa, mas circunscreve a sua abrangência e define o merecido valor para os resultados encontrados.

# Capítulo 8

## Discussão e Conclusão

*Este capítulo encerra esta dissertação com uma breve discussão e conclusão descritas de forma objetiva. Complementam as discussões e conclusões deste capítulo as discussões sobre os resultados da pesquisa experimental (Seção 7.6). A Seção 8.1 discorre sobre as principais contribuições deste trabalho. As conclusões, consequências das conclusões e trabalhos futuros são abordados, em sequência, nas Seções 8.2 e 8.3.*

## 8.1 – Contribuições deste trabalho

Tendo origem na década de 1960 (Papert, 1993), a modelagem computacional com objetivos educacionais continua sendo uma área ativa de pesquisa nos dias atuais e tem sido valiosa para na prática escolar. Paralelamente, sistemas computacionais para computação de raciocínios abduativos têm sido desenvolvidos e estudados em um contexto não educacional. Este trabalho combina a área de pesquisa sobre modelagem computacional em educação com a área de pesquisa sobre sistemas computacionais para abdução, para conceber uma nova modalidade de modelagem denominada de modelagem computacional educacional abduativa.

Sinteticamente, são contribuições deste trabalho a:

- (1) Proposta de diretrizes para modelagem computacional educacional abduativa.
- (2) Instanciação das diretrizes propostas para consubstanciar um ambiente operacional para modelagem computacional educacional abduativa.
- (3) Avaliação do ambiente instanciado.

As diretrizes para modelagem computacional educacional abduativa proposto atua como um guia que orienta os atores (estudantes e mediadores) em relação às atividades que ocorrem dentro do ambiente. Destacam-se nestas diretrizes as orientações sobre as decisões que o estudante deve tomar diante das diferentes situações que podem surgir da avaliação dos resultados da execução de raciocínios abduativos (itens 4 e 5 da Definição 5.1). Não é habitual concepções de ambiente para modelagem computacional oferecerem esta funcionalidade aos atores do ambiente. As diretrizes propostas são gerais e podem ser instanciadas de diferentes maneiras. Por exemplo, pode-se utilizá-las para organizar um ambiente educacional para modelagem abduativa utilizando, digamos, estações de trabalho executando sistemas como Abd1 (Oliveira, Oliveira, Martins e Matsumoto, 2016) ou ACLP (Kakas, Michael & Mourlas, 2000).

A instanciação das diretrizes propostas ocorreram neste trabalho por meio de um ambiente constituído de estações de trabalho equipadas com o sistema SWI-Prolog executando linguagem CHR, estudantes e um mediador. Esta instanciação é também uma contribuição prática deste trabalho porque pode inspirar ou motivar a organização

de ambientes escolares similares ao que foi proposto. Em especial, os exemplos abordados nos capítulos 2, 3 e no Apêndice I podem constituir uma boa fonte de estímulo e inspiração para isto.

Por fim, a avaliação do ambiente instanciado oferece contribuições sobre o que funcionou bem e sobre os desafios técnicos e pedagógicos que devem ser superados para a prática escolar desta modalidade de modelagem, da maneira como foi instanciada, tendo em vista a população estudada.

## **8.2 – Conclusões**

Para a população estudada, os resultados sugerem a viabilidade das diretrizes e do ambiente de modelagem propostos. No entanto, destaca-se que o desempenho dos estudantes nas diferentes habilidades necessárias para a modelagem computacional educacional abduativa são significativamente distintos. Os resultados mostraram que, das quatro habilidades investigadas, a habilidade para abstrair conceitos e a habilidade para avaliar os resultados da execução de modelos são aquelas em que o desempenho dos participantes foi pior, especialmente na primeira. Isto alerta para a necessidade de se tratar tecnicamente e/ou pedagogicamente o ambiente de modelagem para oferecer ao estudante ajuda para a superação das dificuldades impostas à prática destas habilidades. Isto será abordado na próxima seção.

Cabe ressaltar que não há garantia que os resultados deste trabalho possam ser reproduzidos em outras populações com diferentes níveis de conhecimentos e interesses, mas eles criam uma expectativa positiva, devido ao bom desempenho da população estudada nas atividades propostas. Esta expectativa positiva é ampliada ao se considerar que os participantes da pesquisa experimental demonstraram entusiasmo e afetividade com as atividades de modelagem desenvolvidas, favorecendo o estabelecimento de um ambiente de aprendizagem proveitoso.

### **8.3 – Consequências das conclusões e trabalhos futuros**

Os resultados indicam importantes desafios técnicos e pedagógicos a serem superados em relação à abstração de conceitos e à avaliação de resultados da execução de modelos. Dificuldades para abstrair conceitos e para avaliar os resultados da execução de modelos criam barreiras para a continuidade do ciclo de modelagem que, por consequência, afeta negativamente o processo de aprendizagem via atividade de modelagem. Estas dificuldades podem ser amenizadas com medidas de natureza técnica e pedagógica. Trabalhos futuros de natureza pedagógica, como treinamentos sobre a interpretação de resultados, podem contribuir para diminuir as dificuldades de abstrair conceitos ou para atenuar os obstáculos impostos pela linguagem de apresentação dos resultados do sistema SWI-Prolog, usado no ambiente de modelagem proposto. Trabalhos de natureza técnica podem conduzir ao desenvolvimento de uma linguagem para apresentação de resultados mais simples de para ser aprendida e usada.

# Apêndice I

## Formulário: pesquisa experimental sobre modelagem computacional educacional abdutiva

*Este apêndice apresenta integralmente o formulário que foi utilizado na pesquisa experimental sobre Modelagem Computacional Educacional Abdutiva. O formulário está dividido em cinco partes. A Parte 1 declara para o participante o propósito do experimento e permite que ele declare o seu perfil, i.e., sexo, idade, curso e turma. A Parte 2 testa a habilidade da abstração de conceitos. A Parte 3 contém um exercício para ser realizado pelos participantes. O exercício é a proposta de um modelo para ser escrito usando CHR. A Parte 4 testa a habilidade da proposta de fatos para execução de modelos. Por fim, a Parte 5 testa a habilidade da avaliação os resultados da execução de modelos.*

**Faculdade Campo Limpo Paulista**  
**Programa de Mestrado em Ciência da Computação**  
**Pesquisa Experimental sobre Modelagem Computacional Educacional Abdutiva**  
**Parte 1**

**Propósito da Pesquisa e Perfil do Participante**

Código: \_\_\_\_\_

**Propósito da pesquisa**

Modelagem Computacional Educacional é uma atividade de criação de programas que simulam em computador um fenômeno com o objetivo de um estudante aprender sobre o fenômeno. Modelos abduativos são modelos para raciocínios que tem por objetivo formular possíveis hipóteses para explicar fatos observados. Esta pesquisa objetiva investigar facilidades e dificuldades na atividade de modelagem computacional educacional abdutiva.

**Perfil do participante**

1) Sexo:  Masculino  Feminino

2) Idade: \_\_\_\_\_ anos.

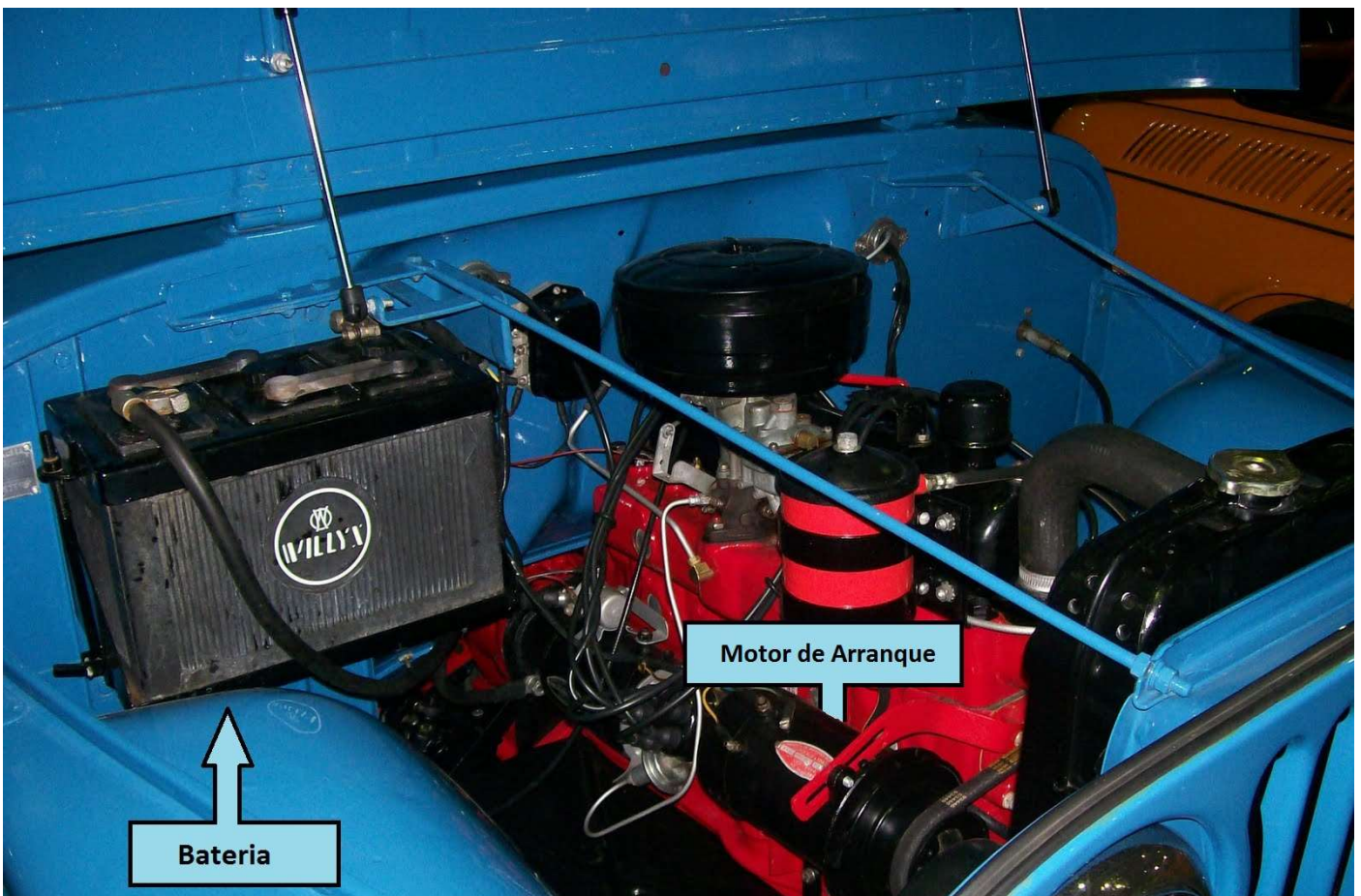
3) Curso: \_\_\_\_\_

4) Turma: \_\_\_\_\_

## Parte 2

### Habilidade para Abstrair Conceitos

Código: \_\_\_\_\_



Considere que a sua intenção é estabelecer um modelo que explique por que o farol esquerdo e/ou o farol direito do Jeep estão apagados ou acesos.

Descreva termos (palavras ou pequenas frases) que você usaria no seu modelo.

- 5) \_\_\_\_\_
- 6) \_\_\_\_\_
- 7) \_\_\_\_\_
- 8) \_\_\_\_\_
- 9) \_\_\_\_\_
- 10) \_\_\_\_\_
- 11) \_\_\_\_\_
- 12) \_\_\_\_\_
- 13) \_\_\_\_\_
- 14) \_\_\_\_\_

Tente relacionar os termos usando conjunções tais como “e” e “ou” de tal modo que explique por que o farol esquerdo e/ou o farol direito do Jeep estão apagados ou acesos. Você pode elaborar uma ou mais frases.

Farol esquerdo do Jeep está apagado:

- 15) \_\_\_\_\_
- 16) \_\_\_\_\_
- 17) \_\_\_\_\_

Farol direito do Jeep está apagado:

- 18) \_\_\_\_\_
- 19) \_\_\_\_\_
- 20) \_\_\_\_\_

Farol esquerdo do Jeep está aceso:

- 21) \_\_\_\_\_
- 22) \_\_\_\_\_
- 23) \_\_\_\_\_

Farol direito do Jeep está aceso:

- 24) \_\_\_\_\_
- 25) \_\_\_\_\_
- 26) \_\_\_\_\_

Considere que a sua intenção é estabelecer um modelo que explique por que o motor do Jeep não dá partida ou dá partida.

Descreva termos (palavras ou pequenas frases) que você usaria no seu modelo.

27) \_\_\_\_\_

28) \_\_\_\_\_

29) \_\_\_\_\_

30) \_\_\_\_\_

31) \_\_\_\_\_

32) \_\_\_\_\_

33) \_\_\_\_\_

34) \_\_\_\_\_

35) \_\_\_\_\_

36) \_\_\_\_\_

Tente relacionar os termos usando conjunções tais como “e” e “ou” de tal modo que explique por que o motor do Jeep não dá partida ou dá partida. Você pode elaborar uma ou mais frases.

Motor do Jeep não dá partida:

37) \_\_\_\_\_

38) \_\_\_\_\_

39) \_\_\_\_\_

Motor do Jeep dá partida:

40) \_\_\_\_\_

41) \_\_\_\_\_

42) \_\_\_\_\_

### Parte 3

#### Habilidade para Descrever um Modelo Usando a Linguagem CHR

Código: \_\_\_\_\_

43) Usando um computador, desenvolva em linguagem CHR um modelo que explique por que o farol esquerdo e/ou o farol direito do Jeep estão apagados ou acesos. Você deve utilizar os termos e as relações expressas a seguir, que informam, em cada caso, por que os faróis esquerdo e direito do Jeep estão apagados e/ou acesos:

farolEsquerdoApagado:

bateriaSemCarga **ou** botaoFarolDesligado **ou** farolEsquerdoQueimado.

farolDireitoApagado:

bateriaSemCarga **ou** botaoFarolDesligado **ou** farolDireitoQueimado.

farolDireitoAceso:

bateriaComCarga **e** botaoFarolLigado **e** farolDireitoOK.

farolEsquerdoAceso:

bateriaComCarga **e** botaoFarolLigado **e** farolEsquerdoOK.

Para facilitar o seu trabalho utilize o arquivo “MeuModelo.pl”, gravado na área de Trabalho. Já está editado no arquivo o cabeçalho necessário para você desenvolver o seu modelo.

## Parte 4

### Habilidade para Propor Fatos para Execução de Modelos

Código: \_\_\_\_\_

Você recebeu um arquivo com um modelo já pronto, denominado “FaroisJeep.pl”, que explica o fato dos faróis do Jeep estarem apagados ou acesos. Com o modelo aberto em um editor, para que você possa ver quais termos foram usados, responda aos exercícios a seguir.

44) Que fatos você deve usar para computar hipóteses para explicar porque o farol esquerdo do Jeep está apagado?

---

---

46) Que fatos você deve usar para computar hipóteses para explica porque o farol esquerdo do Jeep está aceso e o farol direito do Jeep está apagado?

---

---

48) Que fatos você deve usar para computar hipóteses para explicar porque os faróis estão acesos e a bateria está sem carga?

---

---

50) Que fatos você deve usar para computar hipóteses para explicar porque a bateria está sem carga?

---

---

## Parte 5

### Habilidade para Avaliar os Resultados da Execução de Modelos

Código: \_\_\_\_\_

Nos exercícios a seguir escreva, em português, a sua interpretação sobre resultados da execução do modelo “FaroisJeep.pl”:

45) Usando os fatos que você respondeu no exercício 44:

---

---

---

---

---

---

---

---

47) Usando os fatos que você respondeu no exercício 46:

---

---

---

---

---

---

---

---

49) Usando os fatos que você respondeu no exercício 48:

---

---

---

---

---

---

---

51) Usando os fatos que você respondeu no exercício 50:

---

---

---

---

---

---

---

## Parte 5

### Habilidade para Avaliar os Resultados da Execução de um Modelo

No que é pedido a seguir, são descritas várias solicitações de execução de modelos (linha com o sinal de interrogação) e resultados das execuções (as demais linhas). Assinale uma, e apenas uma, alternativa correta em cada uma das questões a seguir.

52) Considerando que ocorreu:

? – fato.

false.

Então pode-se afirmar que:

- a) O fato é falso.
- b) O modelo não possui hipóteses que explicam o fato.
- c) O fato não pertence ao modelo.
- d) O fato, associado a outro fato, se torna verdadeiro.

53) Considerando que ocorreu:

? – fato1, fato2.

fato1

fato2

true;

false.

Então pode-se afirmar que:

- a) O fato1 é verdadeiro e o fato2 é falso.
- b) O fato2 é verdadeiro e o fato1 é falso.
- c) As hipóteses computadas são os fatos apresentados.
- d) O fato1 ou o fato2 são possíveis explicações.

54) Considere a conhecida estória da Chapeuzinho Vermelho e o raciocínio realizado pela personagem Chapeuzinho Vermelho após colecionar fatos em resposta às questões que ela formula no diálogo na casa da Vovó: (1) por que você está com olhos grandes?, (2) por que você está com o nariz tão grande?, (3) por que você está com as orelhas tão grande? e (4) por que a sua boca está tão grande? Considerando que ocorreu:

? – paraMelhorTeVer, paraMelhorTeCheirar, paraMelhorTeOuvir, paraTeDevorar.

lobo

true;

false.

Então pode-se afirmar que:

- a) A hipótese computada pela personagem Chapeuzinho Vermelho é coerente com a estória, considerando o modelo de fantasia da estória.
- b) A hipótese computada pela personagem Chapeuzinho Vermelho é coerente com um modelo cuja intenção é imitar a realidade.
- c) A hipótese computada pela personagem Chapeuzinho Vermelho é parcialmente coerente com a estória, considerando o modelo de fantasia da estória.
- d) A personagem Chapeuzinho Vermelho não computa hipótese sobre os fatos apresentados.

55) Considerando que ocorreu:

? – fato.

a

b

true;

false.

Então pode-se afirmar que:

- a) A execução não gerou hipóteses.
- b) A execução gerou duas hipóteses significando “a ou b”.
- c) A execução gerou duas hipóteses significando “a e b”.
- d) A execução gerou uma hipótese verdadeira e uma falsa.

56) Considerando que ocorreu:

? – fato.

a

b

true;

c

true;

false.

Isto significa que:

- a) A execução não gerou hipóteses.
- b) A execução gerou duas hipóteses significando “( a e b ) ou c”.
- c) A execução gerou uma hipótese significando “a e b e c”.
- d) A execução gerou uma hipótese significando “a ou b ou c”.

# **Apêndice II**

## **Padrão de Resposta aos Exercícios da Pesquisa Experimental**

*Este apêndice apresenta integralmente o formulário que foi utilizado na pesquisa experimental, com as respectivas respostas, para posterior correção e análise dos dados.*

**Faculdade Campo Limpo Paulista**  
**Programa de Mestrado em Ciência da Computação**  
**Pesquisa Experimental sobre Modelagem Computacional Educacional Abdutiva**  
**Parte 1**

**Propósito da Pesquisa e Perfil do Participante**

Código: \_\_\_\_\_

**Propósito da pesquisa**

Modelagem Computacional Educacional é uma atividade de criação de programas que simulam em computador um fenômeno com o objetivo de um estudante aprender sobre o fenômeno. Modelos abduativos são modelos para raciocínios que tem por objetivo formular possíveis hipóteses para explicar fatos observados. Esta pesquisa objetiva investigar facilidades e dificuldades na atividade de modelagem computacional educacional abdutiva.

**Perfil do participante**

1) Sexo:  Masculino  Feminino

2) Idade: \_\_\_\_\_ anos.

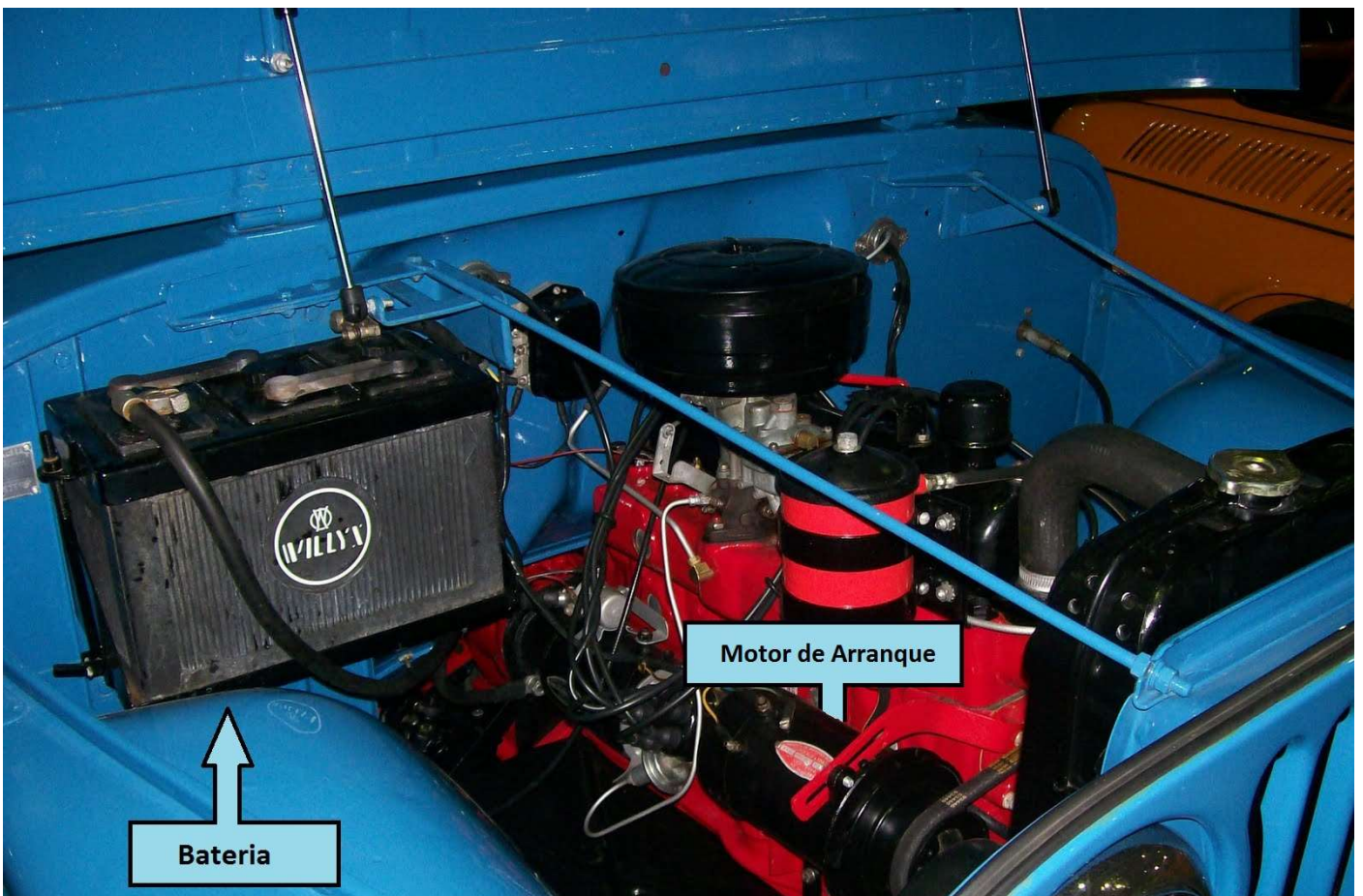
3) Curso: \_\_\_\_\_

4) Turma: \_\_\_\_\_

## Parte 2

### Habilidade para Abstrair Conceitos

Código: \_\_\_\_\_



Considere que a sua intenção é estabelecer um modelo que explique por que o farol esquerdo e/ou o farol direito do Jeep estão apagados ou acesos.

Descreva termos (palavras ou pequenas frases) que você usaria no seu modelo.

- 5) *Bateria sem carga.*
- 6) *Bateria com carga.*
- 7) *Botão do farol desligado.*
- 8) *Botão do farol ligado.*
- 9) *Farol esquerdo queimado.*
- 10) *Farol esquerdo OK.*
- 11) *Farol direito queimado.*
- 12) *Farol direito OK.*

13) \_\_\_\_\_

14) \_\_\_\_\_

Tente relacionar os termos usando conjunções tais como “e” e “ou” de tal modo que explique por que o farol esquerdo e/ou o farol direito do Jeep estão apagados ou acesos. Você pode elaborar uma ou mais frases.

Farol esquerdo do Jeep está apagado:

15) *Bateria sem carga ou botão do farol desligado ou farol esquerdo queimado.*

16) \_\_\_\_\_

17) \_\_\_\_\_

Farol direito do Jeep está apagado:

18) *Bateria sem carga ou botão do farol desligado ou farol direito queimado.*

19) \_\_\_\_\_

20) \_\_\_\_\_

Farol esquerdo do Jeep está aceso:

21) *Bateria com carga e botão do farol ligado e farol esquerdo OK.*

22) \_\_\_\_\_

23) \_\_\_\_\_

Farol direito do Jeep está aceso:

24) *Bateria com carga e botão do farol ligado e farol direito OK.*

25) \_\_\_\_\_

26) \_\_\_\_\_

Considere que a sua intenção é estabelecer um modelo que explique por que o motor do Jeep não dá partida ou dá partida.

Descreva termos (palavras ou pequenas frases) que você usaria no seu modelo.

27) *Bateria sem carga.*

28) *Bateria com carga.*

29) *Motor de arranque com defeito.*

30) *Motor de arranque sem defeito.*

31) *Com falhas de alimentação.*

32) *Sem falhas de alimentação.*

33) \_\_\_\_\_

34) \_\_\_\_\_

35) \_\_\_\_\_

36) \_\_\_\_\_

Tente relacionar os termos usando conjunções tais como “e” e “ou” de tal modo que explique por que o motor do Jeep não dá partida ou dá partida. Você pode elaborar uma ou mais frases.

Motor do Jeep não dá partida:

37) *Bateria s/ carga ou motor de arranque c/ defeito ou c/ falha de alimentação.*

38) \_\_\_\_\_

39) \_\_\_\_\_

Motor do Jeep dá partida:

40) *Bateria c/ carga e motor de arranque c/ defeito e s/ falha de alimentação.*

41) \_\_\_\_\_

42) \_\_\_\_\_

## Parte 3

### Habilidade para Descrever um Modelo Usando a Linguagem CHR

Código: \_\_\_\_\_

43) Usando um computador, desenvolva em linguagem CHR um modelo que explique por que o farol esquerdo e/ou o farol direito do Jeep estão apagados ou acesos. Você deve utilizar os termos e as relações expressas a seguir, que informam, em cada caso, por que os faróis esquerdo e direito do Jeep estão apagados e/ou acesos:

farolEsquerdoApagado:

bateriaSemCarga **ou** botaoFarolDesligado **ou** farolEsquerdoQueimado.

farolDireitoApagado:

bateriaSemCarga **ou** botaoFarolDesligado **ou** farolDireitoQueimado.

farolDireitoAceso:

bateriaComCarga **e** botaoFarolLigado **e** farolDireitoOK.

farolEsquerdoAceso:

bateriaComCarga **e** botaoFarolLigado **e** farolEsquerdoOK.

Para facilitar o seu trabalho utilize o arquivo “MeuModelo.pl”, gravado na área de Trabalho. Já está editado no arquivo o cabeçalho necessário para você desenvolver o seu modelo.

```
:- use_module(library(chr)).
:- chr_constraint farolDireitoApagado, farolEsquerdoApagado,
  botaoFarolDesligado, bateriaSemCarga, farolDireitoQueimado,
  farolEsquerdoQueimado, farolDireitoAceso, farolEsquerdoAceso,
  botaoFarolLigado, bateriaComCarga, farolDireitoOK,
  farolEsquerdoOK.
```

```
bateriaSemCarga \ bateriaSemCarga <=> true.
botaoFarolDesligado \ botaoFarolDesligado <=> true.
farolEsquerdoQueimado \ farolEsquerdoQueimado <=> true.
farolDireitoQueimado \ farolDireitoQueimado <=> true.
bateriaComCarga \ bateriaComCarga <=> true.
botaoFarolLigado \ botaoFarolLigado <=> true.
farolEsquerdoOK \ farolEsquerdoOK <=> true.
farolDireitoOK \ farolDireitoOK <=> true.
farolEsquerdoApagado \ farolEsquerdoApagado <=> true.
farolDireitoApagado \ farolDireitoApagado <=> true.
farolEsquerdoAceso \ farolEsquerdoAceso <=> true.
farolDireitoAceso \ farolDireitoAceso <=> true.
```

```
bateriaSemCarga , bateriaComCarga <=> fail.
botaoFarolDesligado , botaoFarolLigado <=> fail.
farolEsquerdoQueimado , farolEsquerdoOK <=> fail.
farolDireitoQueimado , farolDireitoOK <=> fail.
farolEsquerdoApagado , farolEsquerdoAceso <=> fail.
farolDireitoApagado , farolDireitoAceso <=> fail.
```

```
farolEsquerdoApagado <=> bateriaSemCarga ; botaoFarolDesligado ; farolEsquerdoQueimado.
farolDireitoApagado <=> bateriaSemCarga ; botaoFarolDesligado ; farolDireitoQueimado.
farolDireitoAceso <=> bateriaComCarga , botaoFarolLigado , farolDireitoOK.
farolEsquerdoAceso <=> bateriaComCarga , botaoFarolLigado , farolEsquerdoOK.
```

## Parte 4

### Habilidade para Propor Fatos para Execução de Modelos

Código: \_\_\_\_\_

Você recebeu um arquivo com um modelo já pronto, denominado “FaroisJeep.pl”, que explica o fato dos faróis do Jeep estarem apagados ou acesos. Com o modelo aberto em um editor, para que você possa ver quais termos foram usados, responda aos exercícios a seguir.

44) Que fatos você deve usar para computar hipóteses para explicar porque o farol esquerdo do Jeep está apagado?

farolEsquerdoApagado.

---

46) Que fatos você deve usar para computar hipóteses para explica porque o farol esquerdo do Jeep está aceso e o farol direito do Jeep está apagado?

farolEsquerdoAceso, farolDireitoApagado.

---

48) Que fatos você deve usar para computar hipóteses para explicar porque os faróis estão acesos e a bateria está sem carga?

farolDireitoAceso , farolEsquerdoAceso , bateriaSemCarga.

---

50) Que fatos você deve usar para computar hipóteses para explicar porque a bateria está sem carga?

bateriaSemCarga.

---

## Parte 5

### Habilidade para Avaliar os Resultados da Execução de Modelos

Código: \_\_\_\_\_

Nos exercícios a seguir escreva, em português, a sua interpretação sobre resultados da execução do modelo “FaroisJeep.pl”:

45) Usando os fatos que você respondeu no exercício 44:

*Bateria sem carga ou botão do farol desligado ou farol esquerdo queimado.*

---

---

---

---

---

47) Usando os fatos que você respondeu no exercício 46:

*Bateria com carga e botão do farol ligado e farol esquerdo OK e farol direito queimado.*

---

---

---

---

49) Usando os fatos que você respondeu no exercício 48:

*De acordo com o modelo, não existem hipóteses que explicam os faróis estarem acesos e a bateria estar sem carga.*

---

---

---

---

51) Usando os fatos que você respondeu no exercício 50:

*A hipótese computada é o próprio fato apresentado.*

---

---

---

---

## Parte 5

### Habilidade para Avaliar os Resultados da Execução de um Modelo

No que é pedido a seguir, são descritas várias solicitações de execução de modelos (linha com o sinal de interrogação) e resultados das execuções (as demais linhas). Assinale uma, e apenas uma, alternativa correta em cada uma das questões a seguir.

52) Considerando que ocorreu:

? – fato.

false.

Então pode-se afirmar que:

a) O fato é falso.

x) O modelo não possui hipóteses que explicam o fato.

c) O fato não pertence ao modelo.

d) O fato, associado a outro fato, se torna verdadeiro.

53) Considerando que ocorreu:

? – fato1, fato2.

fato1

fato2

true;

false.

Então pode-se afirmar que:

a) O fato1 é verdadeiro e o fato2 é falso.

b) O fato2 é verdadeiro e o fato1 é falso.

x) As hipóteses computadas são os fatos apresentados.

d) O fato1 ou o fato2 são possíveis explicações.

54) Considere a conhecida estória da Chapeuzinho Vermelho e o raciocínio realizado pela personagem Chapeuzinho Vermelho após colecionar fatos em resposta às questões que ela formula no diálogo na casa da Vovó: (1) por que você está com olhos grandes?, (2) por que você está com o nariz tão grande?, (3) por que você está com as orelhas tão grande? e (4) por que a sua boca está tão grande? Considerando que ocorreu:

? – paraMelhorTeVer, paraMelhorTeCheirar, paraMelhorTeOuvir, paraTeDevorar.

lobo

true;

false.

Então pode-se afirmar que:

x) A hipótese computada pela personagem Chapeuzinho Vermelho é coerente com a estória, considerando o modelo de fantasia da estória.

b) A hipótese computada pela personagem Chapeuzinho Vermelho é coerente com um modelo cuja intenção é imitar a realidade.

c) A hipótese computada pela personagem Chapeuzinho Vermelho é parcialmente coerente com a estória, considerando o modelo de fantasia da estória.

d) A personagem Chapeuzinho Vermelho não computa hipótese sobre os fatos apresentados.

55) Considerando que ocorreu:

? – fato.

a

b

true;

false.

Então pode-se afirmar que:

- a) A execução não gerou hipóteses.
- b) A execução gerou duas hipóteses significando “a ou b”.
- x) A execução gerou duas hipóteses significando “a e b”.
- d) A execução gerou uma hipótese verdadeira e uma falsa.

56) Considerando que ocorreu:

? – fato.

a

b

true;

c

true;

false.

Isto significa que:

- a) A execução não gerou hipóteses.
- x) A execução gerou duas hipóteses significando “( a e b ) ou c”.
- c) A execução gerou uma hipótese significando “a e b e c”.
- d) A execução gerou uma hipótese significando “a ou b ou c”.

# Apêndice III

## Artigos publicados e em processo de avaliação

Durante a realização deste trabalho o autor participou nos seguintes artigos:

- Martins & Oliveira (2015): este artigo foi publicado no WCF 2015, XI *Workshop de Computação na Faccamp*, e refere-se aos primeiros estudos feitos com a linguagem CHR, a fim de conhecê-la. O arquivo aborda como a linguagem CHR pode ser empregada na programação de raciocínios abduativos e conjectura as principais dificuldades do uso desta linguagem por não especialistas em programação lógica.
- Oliveira, Oliveira, Martins & Matsumoto (2016): este artigo foi publicado no MICA I 2016, 15<sup>th</sup> *Mexican International Conference on Artificial Intelligence*, e tem como foco a linguagem Abdl. Foi neste artigo que foram publicadas as primeiras noções daquilo que é descrito, neste trabalho, como as diretrizes para modelagem computacional educacional abduativa.
- Oliveira & Martins (2017): este artigo será publicado no IEEE FIE 2017, *47th Annual Frontiers in Education Conference*, O artigo tem foco na pesquisa experimental sobre a atividade de modelagem educacional abduativa, descrita no Capítulo 7.

Por fim, pretende-se escrever um artigo com as contribuições gerais deste trabalho, com destaque para as diretrizes de modelagem computacional educacional abduativa. O veículo a que se pretende submeter tal artigo ainda não foi definido.

# Referências

- Abdennadher, S. & Christiansen, H. (2000) An experimental CLP platform for integrity constraints and abduction. In: *Proceedings of the Flexible Query Answering Systems: Advances in Soft Computing Series*, FQAS 2000, pp. 141–152.
- Arduino-Org (2017) *Arduino Organization*. Consultado em Abril de 2017. [on-line em abril de 2017]. Em: <https://www.arduino.cc/>.
- Bramer, M. (2013) *Logic programming with Prolog*, 2<sup>nd</sup> ed. London: Springer.
- Campbell T. & Oh P. (2015) Engaging students in modeling as an epistemic practice of science. *Journal of Science Education and Technology*, vol. 24, n. 2, 2015, pp. 125–131.
- Caroprese, L. & Trubitsyna, I. (2014) *A Measure of arbitrariness in abductive explanations*. *Theory and practice of logic programming*, vol. 14, n. 4-5.
- Downey, A. B. (2011) *Physical modeling in MATLAB*. Needham: Green Tea Press.
- Duck, G. J., Stuckey, P. J., Banda, M. J. G. & Holzbaur, C. (2004) The refined operational semantics of Constraint Handling Rules. In: *Proceedings of the 20<sup>th</sup> International Conference on Logic Programming*, ICLP 2004, pp. 90–104.
- Frühwirth, T. (1998) Theory and practice of Constraint Handling Rules. *Journal of Logic Programming*, vol. 37, n. 1-3, pp. 95–138.
- Frühwirth, T. (2008) Welcome to Constraint Handling Rules. *Lecture Notes in Artificial Intelligence*, LNAI 5388. Heidelberg: Springer-Verlag, pp. 1–15.
- Frühwirth, T. (2009). *Constraint Handling Rules*. Cambridge: Cambridge University Press.
- Gavanelli M., Lamma E., Riguzzi F., Bellodi E., Zese R. & Cota G. (2015) An abductive framework for datalog  $\pm$  ontologies. In: *Proceedings of the Technical Communications of the 31st International Conference on Logic Programming*, vol. 1433 of CEUR Workshop Proceedings. CEUR-WS.org.

- Helikar T., Cutucache C. E., Dahlquist L. M., Herek T. A., Larson J. J. & Rogers J. A. (2015) Integrating interactive computational modeling in biology curricula *PLOS computational biology* vol. 11, n. 3, e1004131.
- Jaffar J. & Maher M. J. (1994) Constraint logic programming: a survey. *Journal of Logic Program*, vol. 19, n. 20, pp. 503–581.
- Josephson J. R. & Josephson S. G. (1994) *Abductive inference: computation, philosophy, technology*. Cambridge: Cambridge University Press.
- Kakas, A. C., Kowalski, R. A., & Toni, F. (1993) Abductive Logic Programming. *Journal of Logic and Computation*, vol. 2, n. 6, pp. 719–770.
- Kakas, A. C., Michael, A. & Mourlas, C. (2000) ACLP: Abductive Constraint Logic Programming. *The Journal of Logic Programming*, vol. 44, n. 1-3, pp. 129–177.
- Lazar J., Feng J. H. & Hochheiser H. (2010) *Research methods in human-computer interaction*. Chichester: John Wiley.
- LEGO (2017) LEGO Group. Consultado em Abril de 2017. [on-line em abril de 2017]. Em: <https://www.lego.com>.
- Lopez V. & Hernandez M. I. (2015) *Scratch as a computational modelling tool for teaching physics* *Physics Education*, vol. 50, n. 3, pp. 310–316.
- Magnani, L. (2009) Abductive Cognition: the epistemological and eco-cognitive dimensions of hypothetical reasoning. *Cognitive systems monographs*, vol. 3. Berlin: Springer-Verlag.
- Martins, R. J. & Oliveira, O. L. (2015) Conjecturas sobre dificuldades do uso da Linguagem CHR na programação de raciocínios abduativos por não especialistas em programação lógica. Em: *Anais do XI Workshop de Computação na Faccamp, WCF 2015, ISSN 2446-6808*, vol. 2, pp. 96–100.
- Martins, R. J. & Oliveira, O. L. (2017) Computational modeling of abductive reasoning: an experimental study employing the CHR language in an educational context. In: *Proceedings of 47<sup>th</sup> Annual Frontiers in Education Conference, FIE 2017*, Indianapolis, Indiana, USA.

- McClave, J. T. & Sincich, T. T. (2014) *Statistics*, 12<sup>th</sup> ed. London: Pearson Education.
- Miller, R. S., Ogborn, J., Turner, J., Briggs, J. H. & Brough, D. R. (1990). Towards a tool to support semi-quantitative modelling. In *Advanced Research on Computers in Education*. Amsterdam: R. Lewis & S. Otsuki.
- MSWLogo (2017) *MSWLogo*. Consultado em Abril de 2017. [on-line em abril de 2017]. Em: <http://www.softronix.com/logo.html>.
- Nicoletti, M. C. (2017) *A cartilha da lógica*, 3<sup>a</sup>. ed., LTC.
- Oliveira, C. E. A. (2016) *Abd1: um protótipo de linguagem específica para programação de raciocínios abduativos*. Dissertação de Mestrado, Faccamp, [on-line em abril de 2017]. Em: <http://www.cc.faccamp.br/Dissertacoes/CarlosEduardoAndrade.pdf>.
- Oliveira, O. L., Oliveira, C. E. A., Martins, R. J. & Matsumoto, M. (2016) A logic programming language designed for the modeling of abductive reasoning in an educational context. In: *Proceedings of the Mexican International Conference on Artificial Intelligence, MICAI 2016*. Cancún, México, pp. 1–13.
- Papert, S. (1993) *Mindstorms: children, computers, and powerful ideas*, 2<sup>nd</sup> ed. New York: Basic Books.
- Pospel, H. (2003) *Introduction to logic: Predicate Logic*, 2<sup>nd</sup> ed. Pearson.
- Richmond, B., Peterson, S & Vescuso, P. (1987) *An academic user's guide to STELLA*. Lyme: High Performance System, Inc.
- Rodrigues, F., Oliveira, C. E. A. & Oliveira, O. L. (2014) Peirce: an algorithm for abductive reasoning operating with a quaternary reasoning framework, *Research in Computer Science*, vol. 82, pp. 53–66.
- Rodrigues, F., Oliveira, C. E. A. & Oliveira, O. L. (2014) Peirce: an algorithm for abductive reasoning operating with a quaternary reasoning framework. In: *Proceedings of the 13<sup>th</sup> Mexican International Conference on Artificial Intelligence, MICAI 2014*, Tuxtla Gutiérrez, Chiapas, Mexico, 16-22 November, 2014, pp. 55–68.

- Rodrigues, F. (2015) *Um algoritmo para abdução peirceana baseado em uma estrutura de raciocínio quaternária em lógica*, Dissertação de Mestrado, Faccamp, [on-line em abril de 2017]. Em: [http://www.cc.faccamp.br/Dissertacoes/Felipe\\_2015.pdf](http://www.cc.faccamp.br/Dissertacoes/Felipe_2015.pdf).
- Russell, S. & Norvig, P. (2014) *Artificial intelligence. A modern approach*, 3<sup>rd</sup> ed. New Jersey: Prentice Hall, Englewood Cliffs, pp 471–497.
- SWI-Prolog-Org (2017) *SWI-Prolog Organization*. Consultado em Abril de 2017. [on-line em abril de 2017]. Em: <http://www.swi-prolog.org/>.
- Thagard, P. R. (1978) The best explanation: criteria for theory choice. *The Journal of Philosophy*, vol. 75, n. 2, pp. 76–92.
- Thagard, P. R. (1989) *Explanatory coherence. Behavioral and brain sciences*, vol. 12, n. 3, pp. 435–502.
- Wikipédia (2017) *Classificação biológica dos seres vivos*. [on-line em abril de 2017]. Em: [https://pt.wikipedia.org/wiki/Classificação\\_científica](https://pt.wikipedia.org/wiki/Classificação_científica).