

UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE CIÊNCIAS E TECNOLOGIA
COORDENAÇÃO DE PÓS-GRADUAÇÃO EM INFORMÁTICA

**Um Modelo Genérico de Agente Racional Baseado em Raciocínio Assíncrono
Interativo**

Lidiana de França Martins

Campina Grande – PB
Junho - 2002

UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE CIÊNCIAS E TECNOLOGIA
COORDENAÇÃO DE PÓS-GRADUAÇÃO EM INFORMÁTICA

**Um Modelo Genérico de Agente Racional Baseado em Raciocínio Assíncrono
Interativo**

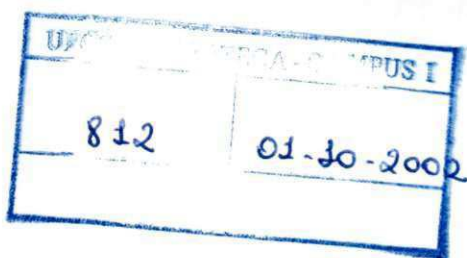
Lidiana de França Martins

Dissertação apresentada ao curso de Mestrado em
Informática do Departamento de Sistemas e
Computação da Universidade Federal da Paraíba,
como requisito para a obtenção do título de Mestre
em Informática.

Orientador: Dr. Edílson Ferneda

Área de atuação: Teoria da Computação e Inteligência
Artificial

Campina Grande – PB
Junho - 2002



Ficha Catalográfica

MARTINS, Lidiana de França

M382M

Um modelo Genérico de Agente Racional Baseado em Raciocínio Assíncrono Interativo

Dissertação (Mestrado) – Universidade Federal da Paraíba – Centro de Ciências e Tecnologia
– Coordenação de Pós-Graduação em Informática, Campina Grande – Pb, Junho de 2002.

79 p. Il.

Orientador: Edilson Farneda

Palavras-chaves:

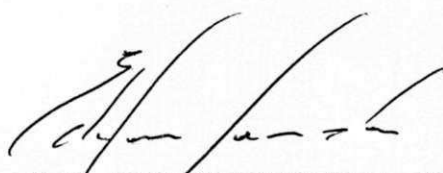
- 1- Inteligência Artificial
- 2- Descoberta Científica
- 3- Agente Racional

CDU – 007.52

**UM MODELO GENÉRICO DE AGENTE RACIONAL BASEADO EM
RACIOCÍNIO ASSÍNCRONO INTERATIVO**

LIDIANA DE FRANÇA MARTINS

DISSERTAÇÃO APROVADA EM 19.06.2002



PROF. EDILSON FERNEDA, Dr.
Orientador



PROF. EVANDRO DE BARROS COSTA, D.Sc
Examinador



PROF. AGENOR DE SOUZA MARTINS, D.Sc
Examinador

CAMPINA GRANDE – PB

À Juliana e Heleno,
meus pais.

Agradecimentos

À Deus, por ter conseguido a façanha de mostrar, ainda aos 25 anos, o que realmente é a vida.

A minha família, por ter apoiado minhas decisões e respeitado os meus sonhos mais ousados.

À Manuela, pela jóia valiosa com a qual me presenteou.

À Yvson, Giulliana, Jack e Eulanda, pela amizade e apoio em momentos tão decisivos.

À Edílson, Germana, Sallantin, Álvaro, Joelson, Carmem e Claudia, comment je peux remercier pour tout qui vous faites pour moi?

Aos meus colegas do LIA, à representação discente e aos funcionários do DSC.

À CAPES, pelo suporte financeiro.

À Agesandro, por ter sido meu amigo, meu namorado, meu conselheiro e muitas outras coisas que eu não sei como explicar.

Resumo

A ferramenta teórica ϕ -calculus é uma tentativa de gerenciar raciocínio assíncrono interativo no contexto das ciências empíricas. Um instrutor (um ser humano especialista do domínio), com conhecimento sobre determinado assunto e interesse em rever e evoluir este conhecimento, teria como colaborador um agente artificial, sistema que aprenderá (e raciocinará sobre) o conhecimento que o instrutor detém. Haja vista a importância de se verificar como o conhecimento evoluiu, após a fase de ensino, duas outras se iniciam: a avaliação do conhecimento (do agente artificial pelo especialista do domínio) e sua correção (quando acontece o refinamento do conhecimento aprendido). Este processo de colaboração/aprendizagem entre o especialista do domínio e o agente artificial é assíncrono e interativo. Inspirada em ϕ -calculus e baseada em raciocínio assíncrono interativo, uma arquitetura é proposta.

Abstract

The φ -calculus theoretical tool tries to manage asynchronous interactive reasoning in the empirical sciences context. An instructor (a human specialist agent of the domain) with knowledge about the subject and interest in reviewing and evolving this knowledge collaborates with artificial agent which learns (and reasons about) the knowledge that the instructor withholds. It is important to verify how such knowledge evolved, so after the learning phase, two other phases initiate: the knowledge evaluation (by the domain specialist) and the correction (which happens after the refining of the learnt knowledge). This collaboration/learning process between the domain specialist and the artificial agent is interactive and asynchronous. An architecture, inspired in φ -calculus and based on interactive asynchronous reasoning, is proposed.

Sumário

1. Introdução	1
1.1 Visão geral.....	1
1.2 Objetivos da dissertação	2
1.3 Contextualização	2
1.4 Estrutura da dissertação	3
2. Descoberta Científica Computacional	4
2.1 Introdução.....	4
2.2 Fundamentos em descoberta científica.....	5
2.3 Descoberta científica e aprendizagem	8
2.4 Sistemas de descoberta científica	9
2.5 Sistemas de construção interativa de teorias	11
2.6 Conclusão	15
3. Abordagens teóricas para a construção interativa de teorias	16
3.1. Introdução	16
3.2 O projeto SAID	17
3.2.1 Teorias Semi-Empíricas.....	18
3.2.2 Esquema mental	20
3.2.3 Um protocolo para aprendizagem.....	20
3.2.4 Sistemas de crenças	22
3.2.5 Princípios para a geração e evolução do conhecimento	23
3.2.6 Discussão	25
3.3 O projeto ϕ -calculus	26
3.3.1 Motivação inicial: a origem do método	27
3.3.2 Fundamentos teóricos	28
3.3.2.1 Expressão	30
3.3.2.2 Formalização	31
3.3.2.3 Computação.....	33
3.3.2.4 Busca por adequação	33
3.3.3 Discussão	36

3.4 Conclusão.....	38
4. Um sistema de construção interativa de teorias	40
4.1 Introdução	40
4.2 O raciocínio assíncrono interativo	41
4.3 O processo de modelagem	43
4.3.1 Vertente conceitual	45
4.3.1.1 Fase de ensino	45
4.3.1.2 Fase de avaliação.....	47
4.3.1.3 Fase de correção	48
4.3.2 Vertente formal	50
4.3.2.1 Fase de ensino	52
4.3.2.2 Fase de avaliação.....	54
4.3.2.3 Fase de correção	55
4.3.3 Vertente experimental.....	57
4.3.3.1 Fase de ensino	58
4.3.3.2 Fase de avaliação.....	59
4.3.3.3 Fase de correção	60
4.3.3.4 Considerações.....	60
4.4 Discussão	62
4.5 Conclusão.....	65
5. Conclusão	67
5.1. Considerações finais	66
5.2. Trabalhos futuros	67
Referências Bibliográficas	69
Anexo A: WCP – um programa demonstrador da metodologia empregada em φ- calculus	73
Anexo B: Autômato Pushdown	78

Lista de figuras

Figura 2.1: As três categorias popperianas	6
Figura 2.2: Heurísticas da descoberta matemática	7
Figura 2.3: A importância do agente humano no processo de descoberta científica vista em seis passos.....	12
Figura 3.1: Mecanismos do agente racional SAID.....	18
Figura 3.2: Termos que intervêm na formalização e na evolução do conhecimento pelas TSE.....	19
Figura 3.3: Protocolo de aprendizagem MOSCA do ponto de vista de resolução de problemas	21
Figura 3.4: Sistema de crença considerado	23
Figura 3.5: O processo de busca por adequação.....	35
Figura 3.6: O cenário de experimentação.....	38
Figura 4.1: Agente aprendiz baseado em raciocínio assíncrono interativo	41
Figura 4.2: A fase de ensino (a) indireto e (b) direto	46
Figura 4.3: A fase de avaliação	47
Figura 4.4: A fase de correção.....	49
Figura 4.5: Ciclo de modelagem segundo os processos da metodologia do ϕ -calculus	50
Figura 4.6: Autômato representado o modelo genérico de agente aprendiz	51
Figura 4.7: Autômato representado a fase de ensino.....	53
Figura 4.8: Autômato representado a fase de avaliação	54
Figura 4.9: Autômato representado a fase de correção	55

Figura 4.10: Amostra de aprendizagem.....	57
Figura 4.11: Termos escolhidos pelo instrutor	57
Figura 4.12: Amostra de avaliação	59
Figura 4.13: Nova amostra de avaliação	61
Figura 4.14: Nova hierarquia de termos	61
Figura 4.15: Modelo genérico de agente aprendiz proposto por Russell e Norvig	62
Figura 4.16: Agente SAID adaptado ao modelo genérico de agente aprendiz de Russell e Norvig	63
Figura 4.17: Agente baseado em raciocínio assíncrono interativo adaptado ao modelo genérico de agente aprendiz de Russell e Norvig	64

Capítulo 1

Introdução

1.1 Visão Geral

Um bom modelo é raramente obtido em um único passo - geralmente é construído progressivamente, por refinamento sucessivo, na medida em que o especialista do domínio confronta a realidade (sendo modelada) com o modelo (sendo construído). De acordo com este pensamento, a evolução bem-sucedida de um modelo é fortemente dependente das habilidades e experiência do especialista do domínio, que decide quando modificá-lo. A tarefa de criação de modelos seria melhor desempenhada se a intervenção humana fosse cuidadosamente considerada nas metodologias, ferramentas e ambientes de modelagem [Nóbrega 02a, 02b]. Modelo é aqui definido como uma imagem reproduzida pelo especialista do domínio acerca do seu próprio conhecimento – esta imagem reproduzida deve ter correspondência com aspectos do mundo real que fazem sentido para o especialista.

Neste contexto, uma teoria pode ser vista como um modelo que verifica uma *adequação*¹ com relação aos experimentos. Tal adequação é julgada sob as qualidades de *predição* e *explicação* da teoria. Na verdade, a teoria deve ser formulada em uma linguagem formal que, por um lado, permita descrever fenômenos e, por outro lado, provenha propriedades de computação. A teoria deve ser ligada – por um mecanismo por indução – aos fatos, hipóteses e situações particulares (chamadas de exemplos) que descrevem a realidade sendo modelada.

Alguns trabalhos desenvolvidos nesta área [Ferneda 92a, 92b; Martins 98; Nóbrega 98; Silva 98; Carvalho 99, 01; Gueyi 00] foram direcionados para a concepção de sistemas agentes racionais capazes de construir teorias sob a supervisão de um especialista – teorias que exibam propriedades de predição e explicação no contexto de aperfeiçoamen-

¹ Adequação refere-se ao julgamento humano com relação às representações do mundo, de acordo com o seu ponto de vista.

to experimental. Os futuros sistemas de apoio à descoberta poderiam ser aperfeiçoados se os projetistas dispusessem de guias que lhes dissessem o que observar durante a construção de tais sistemas. A metodologia ϕ -calculus suporta construção de teorias sob supervisão humana [Sallantin 00, Nóbrega 00, 01a, 01b, 02a, 02b].

1.2 Objetivos da dissertação

ϕ -calculus é uma ferramenta teórica para a construção interativa e racional de teorias [Nóbrega 00, 01a, 01b]. A construção de um modelo dentro de ϕ -calculus é apreendida como um processo interativo, em que a intervenção humana tem um papel decisivo durante a construção/evolução do mesmo. Através da experimentação, em que o modelo é confrontado com um fragmento da realidade, o especialista do domínio dirá se o modelo é adequado ou não, e se precisa de aperfeiçoamento. A adequação se dá em três níveis: ontológico, epistêmico e heurístico (ver capítulo 3 para maiores detalhes).

Este trabalho de dissertação de mestrado visa conceber um modelo genérico de agente aprendiz baseado em raciocínio assíncrono interativo e inspirado em ϕ -calculus. A idéia geral é verificar como acontece toda a interação entre o sistema, também chamado de agente artificial, e o instrutor humano, especialista do domínio responsável pela aprendizagem e com o qual o sistema colabora. Esta interação é analisada segundo três vertentes: conceitual, formal e experimental.

1.3 Contextualização

Os trabalhos dentro do Grupo de Inteligência Artificial (GIA), na área de descoberta científica, têm sido direcionados para a concepção e o desenvolvimento de um sistema interativo de apoio à descoberta científica baseado na noção de agente racionais, o projeto SAID. Apesar de todos os esforços, o SAID possui alguns problemas ainda não solucionados: (a) o instrutor não tem acesso direto ao conhecimento armazenado pelo agente, tanto é que a interação entre eles é feita apenas a partir das argumentações do agente, (b) não existe uma idéia clara acerca do processo de busca por adequação, e (c) o instrutor tem que saber de antemão qual a solução para o problema que ele deseja solucionar a partir desta interação. No modelo genérico de agente racional, proposto neste trabalho, estes problemas foram solucionados: (a) o instrutor pode acessar/modificar/acrescentar diretamente a informação que o agente dispõe, (b) o processo de busca por adequação é melhor definido, e (c) o instrutor não precisa conhecer a priori a solução para o problema. Este modelo é ba-

seado na ferramenta teórica ϕ -calculus e na idéia de raciocínio assíncrono interativo. Como ϕ -calculus ainda não está completamente finalizado, este trabalho foi realizado tomando como base o que conhecido/definido sobre esta ferramenta.

Para maiores detalhes, consultar os capítulos 3 e 4.

1.4 Estrutura da dissertação

A dissertação está estruturada da seguinte forma:

- **Capítulo 1 – Introdução:** contextualização do trabalho, objetivos e relevância – mostrados neste capítulo.
- **Capítulo 2 – Descoberta científica computacional:** o capítulo 2 tem como objetivo conceder uma visão geral acerca da área de conhecimento *Descoberta Científica*. São abordados os fundamentos deste campo de pesquisa, a relação entre descoberta científica e aprendizagem, os sistemas clássicos de descoberta científica e a construção interativa de teorias. Também é discutida a importância da interação instrutor-aprendiz na concepção de teorias por entidades artificiais.
- **Capítulo 3 – Abordagens teóricas para a construção interativa de teorias:** o capítulo tem como objetivo principal definir o que é raciocínio assíncrono interativo. Este capítulo está estruturado em duas seções principais: a primeira, onde será mostrado o agente racional SAID (objeto de estudo de trabalhos anteriores de membros do GIA), e a outra, onde será abordada a ferramenta teórica de concepção interativa ϕ -calculus (estrutura de representação/evolução de conhecimento e busca por adequação). Uma discussão é feita acerca das duas abordagens.
- **Capítulo 4 – Um sistema de construção interativa de construção de teorias:** o sistema proposto é um modelo genérico de agente racional aprendiz, inspirado em ϕ -calculus e baseada em raciocínio assíncrono interativo. É feita uma análise da interação instrutor-agente segundo três vertentes complementares: conceitual, formal e experimental.
- **Capítulo 5 – Conclusão:** considerações e perspectivas acerca do trabalho.
- **Anexo A – Um programa demonstrador da metodologia empregada em ϕ -calculus**
- **Anexo B – Autômato pushdown**

Capítulo 2

Descoberta Científica Computacional

2.1 Introdução

Em descoberta científica, vários sistemas simularam com sucesso descobertas importantes registradas na história da ciência. Exemplos desses sistemas são o Eurisko [Lernat 83b], o Bacon [Langley 81] e o Boole [Ledesma 97]. A concepção de tais sistemas contribuiu fortemente para que se chegasse às bases de uma teoria normativa da descoberta.

Considerando a descoberta modelada como um processo de resolução de problema, como proposto por Newell *et al* [Newell 62], uma teoria normativa da descoberta parte do princípio de que teorias raramente surgem de pesquisas aleatórias, uma vez que isto tenderia a ser impraticável dada a extensão dos espaços de busca. Nesse contexto, uma noção fundamental é a de *racionalidade*. Para um cientista, tal noção consiste em usar o melhor meio possível para limitar os espaços de busca a proporções manipuláveis.

Sistemas de descoberta, contudo, pareciam mostrar um comportamento limitado no que diz respeito à continuidade na evolução das teorias que eles manipulavam. Segundo Simon *et al* [Simon 97], “sistemas de descoberta que resolvam problemas em cooperação com um especialista no domínio desempenham um importante papel, uma vez que, em qualquer domínio não trivial, é virtualmente impossível prover um sistema com uma teoria completa¹, haja vista que ela, em geral, evolui constantemente”. Assim, observamos que, no contexto de descoberta, um sistema que não considera a *interação* com o usuário irá certamente deparar-se com limitações no que concerne a sua capacidade de evoluir continuamente. Dessa forma, ao invés de sistemas de descoberta, uma alternativa seria a dispo-

nibilização de sistemas de apoio à descoberta, incorporando-lhe a habilidade de acompanhar a evolução contínua do conhecimento em um domínio. As vantagens da interação homem-máquina em descoberta científica são evidenciadas por uma série de sistemas, como reportado por Langley [Langley 98, 00], que recomenda que futuros sistemas desse tipo considerem um suporte mais explícito à intervenção humana no processo de descoberta. Outros trabalhos [Jong 97, Valdes-Perez 99] já sugerem algumas diretrizes para a agilização do desenvolvimento de sistemas de apoio à descoberta.

A construção de sistemas de apoio à descoberta já é observada na prática há algum tempo, como ilustrado pelos diversos trabalhos [Ferneda 92a, Nóbrega 98] onde sistemas foram concebidos como agentes racionais [Sallantin 91a, 91b, 97], no sentido de que eles são capazes de construir teorias sob a supervisão de um usuário especialista. Além disso, as teorias que os agentes racionais são capazes de construir em cooperação com um especialista no domínio apresentam as propriedades requeridas por uma teoria científica, tais como *predictibilidade e explicabilidade*.

A área de Descoberta Científica não tem uma definição precisa. Praticamente todos os trabalhos dentro desta área estão direcionados à redescoberta de conhecimentos já conhecidos, bem como a busca de novas teorias “pensadas” por entidades artificiais. Com base nisso, pergunta-se: como se dará a aprendizagem do sistema? Como será a interação entre o instrutor, caso ele exista, e o sistema? Como o sistema representará seu conhecimento? Como este conhecimento evoluirá? Estas são algumas das “preocupações” dos pesquisadores.

Neste capítulo serão abordados os fundamentos teóricos da área de Descoberta Científica, a relação entre Descoberta Científica e Aprendizagem de Máquina, serão apresentados alguns exemplos de sistemas que simulam descoberta e será aprofundado o tema de descoberta de construção interativa de teorias.

2.2 Fundamentos em descoberta científica

A Filosofia e as ciências ocidentais nasceram da busca do homem pelas explicações da natureza e de seus fenômenos. O conhecimento é alcançado através da coleta de fatos com observação organizada, derivando, a partir daí, as teorias [Bacon 84].

¹ Entende-se por teoria completa, neste contexto, como uma teoria concluída - que não precisa ser aperfeiçoada.

Na Lógica da Descoberta Científica de Popper [Popper 93], a teoria científica é obtida pela invenção de hipóteses, conjecturas e até mesmo por “adivinhações” (é importante ressaltar que a teoria deve descrever coisas do mundo real e manter uma correspondência com elas, não importando como foi obtida). Para Popper, uma teoria é científica se satisfaz o *Princípio da Refutabilidade*, ou seja, se é capaz de prever algo sobre o mundo real passível de falsificação. Quanto à progressão da ciência, ela se faria a partir da *Lógica Situacional de Problemas*, que diz respeito à explicação conjectural de alguma ação humana que recorra à situação onde o agente se encontra. Popper estabeleceu uma tese, também conhecida como *as três categorias popperianas*, onde existe um elo de ligação entre três mundos: o *mundo material* (mundo dos estados materiais), o *mundo mental* (mundo dos estados mentais) e o *mundo ideal* (mundos das idéias no sentido objetivo). A mente estabelece um elo indireto entre o primeiro e o terceiro mundo (figura 2.1).

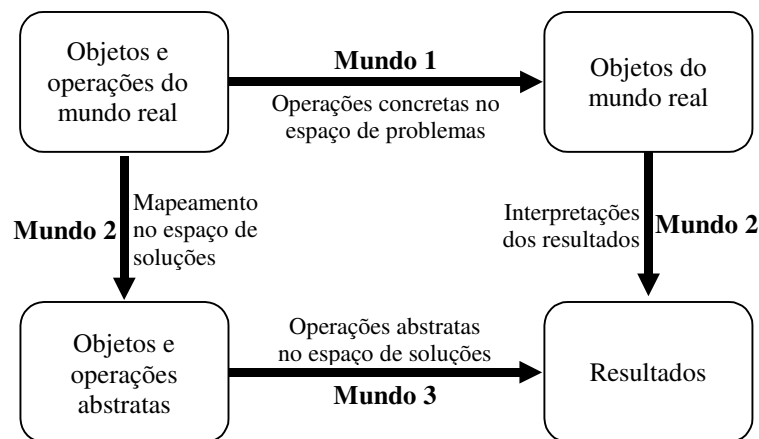


Figura 2.1: As três categorias popperianas.

Na Lógica das Provas e Refutações de Lakatos, o conhecimento matemático é hipotético, conjectural e desenvolvido por meio de especulações e críticas:

“... a matemática não formal, semi-empírica, não progride mediante monótono aumento do número de teoremas indubitavelmente estabelecidos, mas mediante incessante aperfeiçoamento de opiniões por especulação e crítica, pela lógica das provas e refutações” [Lakatos 76].

Seu objetivo é estudar a construção de uma prova e determinar o seu domínio de validade pela análise de exemplos e contra-exemplos desta prova.

As bases da heurística da descoberta matemática são esquematizadas na figura 2.2. A Lógica das Provas e Refutações pode ser resumida da seguinte forma: (*passo 1*) obter uma conjectura; (*passo 2*) elaborar uma prova (experiência mental sobre os exemplos

da conjectura) e enunciar os lemas que constitui a prova analítica inicial; (*passo 3*) exibir os contra-exemplos para a conjectura (aspecto global) e/ou aos lemas suspeitos (aspecto local); e (*passo 4*) construir a prova analítica para retificação de lemas ou para adicionar lemas, que devem também ser introduzidos como condição à conjectura. Devido ao fato do método se interessar tanto pela elaboração de uma prova da conjectura como pela crítica dessa prova, o método das Provas e Refutações exige uma abordagem interativa.

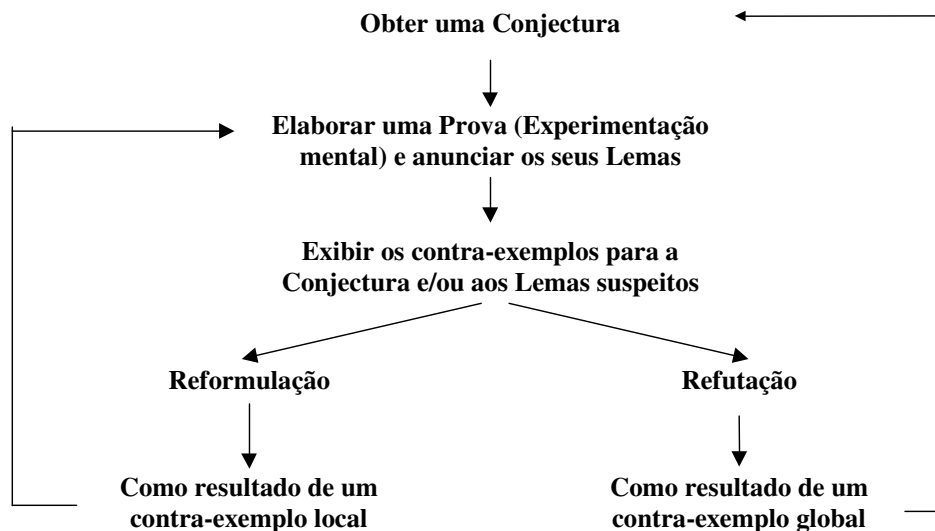


Figura 2.2: Heurística da descoberta matemática.

Por outro lado, dentre os filósofos do século XX, diversos foram os que estavam mais preocupados na verificação da validade do conhecimento científico do que na formulação de uma teoria para a descoberta (que seria apenas produto da criatividade). A tarefa de construir uma teoria da descoberta foi então deixada a cargo dos psicólogos. Newell, Shaw e Simon propuseram que a descoberta poderia ser modelada como um processo de resolução de problemas [Newell 62], sendo a *Racionalidade* importante para limitar o espaço de busca em uma pesquisa.

Por várias décadas, os processos de descoberta de leis e conceitos científicos têm sido um tópico de pesquisa importante dentro da Inteligência Artificial [Simon 97]. Esta pesquisa está direcionada a duas metas principais: (i) compreender o raciocínio científico humano e (ii) desenvolver sistemas que, automaticamente (ou em colaboração com um especialista do domínio), contribuam para a descoberta de novos conhecimentos científicos. As duas metas podem parecer não complementares, mas, de fato, idéias derivadas da pesquisa em descoberta científica humana têm sido aplicadas no projeto de sistemas artificiais, enquanto o projeto de tais sistemas contribui para as idéias acerca dos processos que po-

dem ser usados pelos humanos quando desenvolvem tarefas de descoberta. A interação entre estas duas atividades traz contribuições mútuas.

2.3 Descoberta científica e aprendizagem

A habilidade para aprender é uma das características centrais da inteligência, sendo objeto de estudo para as áreas de Psicologia Cognitiva e Inteligência Artificial [Langley 96]. O campo da *Aprendizagem de Máquina*, que cruza estas duas disciplinas, estuda os processos computacionais que envolvem a aprendizagem de humanos e de máquinas. A meta principal é estabelecer modelos lógicos e matemáticos do raciocínio humano e então utilizá-los em um computador.

Tal como para *inteligência e racionalidade*, também não se tem uma idéia clara sobre o conceito de aprendizagem. Com base nisto, pergunta-se: a habilidade para aprender é exclusiva dos seres humanos? O fato de um animal realizar uma tarefa idealizada por um humano caracteriza uma aprendizagem? [Carvalho 99]. Este processo inclui aquisição, generalização e representação de novos conhecimentos; descoberta de novos fatos/teorias via, por exemplo, observação e experiência.

A pesquisa em *Aprendizagem de Máquina* é importante em diversas situações, tais como (i) quando há procedimentos cuja compreensão se dá através de exemplos, (ii) quando se quer extrair relações existentes escondidas em uma grande quantidade de dados, (iii) quando os projetos precisam progredir junto com o trabalho (as características do ambiente não são conhecidas em tempo de projeto) e (iv) quando a quantidade de conhecimento é tão grande que este não pode ser codificado explicitamente.

Em um processo de aprendizagem, a informação usada pelo sistema é de alguma forma representada para ser inserida na máquina. A qualidade do conhecimento aprendido é então validado com o auxílio de um especialista humano ou com outro componente baseado em conhecimento. Por fim, o aprendiz é submetido a novas situações.

Tendo-se a Descoberta Científica como uma estratégia de Aprendizagem de Máquina, pode-se dizer que o processo de aprendizagem inclui três passos: (i) aquisição de novos conhecimentos, (ii) organização de novos conhecimentos em representações genéricas e (iii) descoberta de novos fatos, teorias, etc. As estratégias são as seguintes [Michalski 83]:

- *Aprendizagem por memorização.* Dá-se pela memorização dos dados que o aprendiz recebe, não havendo nenhuma inferência;
- *Aprendizagem por instrução.* A informação deve ter uma representação que possa ser manipulada diretamente pelo aprendiz. Essa informação é integrada ao seu conhecimento prévio. O aprendiz também faz inferência sobre o conhecimento que possui, mas o instrutor ainda possui uma grande responsabilidade com relação à introdução e organização do conhecimento inicial do aprendiz;
- *Aprendizagem por analogia.* Um sistema que aprende por analogia pode converter um programa existente em um outro que realiza uma tarefa semelhante à original. Isso, porém, demanda muita inferência. Um fato ou habilidade análogos ao que se quer aprender é um parâmetro bastante importante a ser retido pelo aprendiz;
- *Aprendizagem por exemplos.* Como abordado anteriormente, bastante utilizado na IA. Dado um conjunto de exemplos e contra-exemplos de um conceito, o aprendiz induz uma descrição do conceito que descreve os exemplos e não os contra-exemplos. A fonte de informação para o aprendiz pode ser um instrutor, o próprio aprendiz ou o ambiente externo;
- *Aprendizagem por observação e descoberta.* Esta é uma forma de aprendizagem bastante genérica de aprendizagem indutiva que inclui sistemas de descoberta, tarefas de formação de teorias, criação de critérios de classificação para formar hierarquias taxonômicas e tarefas similares sem a intervenção de um instrutor. Requer que o aprendiz realize mais inferência do que todas as outras estratégias.

A *aprendizagem por exemplos* (ou *Aquisição de Conceitos*) é um método indutivo que identifica caracterizações de alguns objetos (situações, processos, etc.) pré-classificados por um instrutor em uma ou mais classes, ou conceitos [Carvalho 99]. A fonte de exemplos pode ser um instrutor (que gera seqüência de exemplos para o aprendiz), o próprio aprendiz (que gera instâncias do que conceito a ser aprendido, utilizando uma base de informações) ou o ambiente externo (o processo de geração dos exemplos é aleatório).

2.4 Sistemas de descoberta científica

A modelagem do processo de descoberta através de um sistema computacional é um passo importante no sentido de uma normatização para esse processo [Nóbrega 98].

Exemplos de sistemas (históricos) de descoberta científica já construídos são:

- **AM (Automatic Mathematician)**. Concebido em 1977 para modelar um aspecto de pesquisa em matemática elementar, descobriu novos conceitos guiado por uma larga gama de heurísticas [Lenat 83a]. Os conceitos são representados por conceitos do tipo *frame*. As heurísticas se comunicam via mecanismo de agenda, uma lista global de tarefas a serem realizadas pelo sistema. Uma tarefa pode levar o AM a definir um novo conceito (um *slot* de um *frame*) para examinar regularidades em dados empíricos. O sistema encontra a tarefa mais plausível e a executa. Há 115 estruturas, cada uma correspondendo a um conceito elementar da teoria dos conjuntos, e 255 heurísticas. Algumas heurísticas são utilizadas para selecionar conceitos ou as propriedades de conceitos a serem explorados, enquanto outras procuram por informações relacionadas a uma propriedade escolhida. Outras ainda identificam relacionamentos entre conceitos já conhecidos. AM descobriu conceitos em Teoria dos Números, em especial os Naturais e os Primos.
- **Eurisko**. No AM, as heurísticas não evoluíam como os conceitos. Eurisko parte do princípio que as heurísticas deveriam ser tratadas como tal [Lenat 83b].
- **Bacon**. Sistema com o propósito de descobrir leis quantitativas. Para tal, baseou-se em um modelo de descoberta científica dirigida por dados [Langley 81]. Recebe um conjunto de variáveis dependentes e independentes e gera leis que relacionam essas variáveis entre si. Bacon redescobriu a *Lei do Gás Ideal*, a *Terceira Lei de Kepler*, a *Lei de Coulomb* e a *Lei de Ohm* [Langley 92].
- **Glauber**. Sistema para a descoberta para as leis qualitativas em Química [Langley 92]. O sistema recebe algumas leis qualitativas, as reações químicas e as substâncias envolvidas. Por fim, Glauber propõe leis mais genéricas. Redescobriu a *Lei dos Ácidos e das Bases*.
- **Boole2**. Sistema que recebe como parâmetro de entrada um conjunto de operações e um conjunto de combinações representando um conhecimento científico. O resultado gerado é um registro das propriedades algébricas do tal conhecimento e uma indicação sobre a possibilidade deste ser passível de sofrer separação de símbolos (*simbolizável*). O programa foi capaz de descobrir que a Lógica, a Geometria e um subconjunto do Cálculo Diferencial são simbolizáveis e que a generalização da Geometria Gregoriana não o é devido ao seu leque de comutatividade. É interessante observar

que, diferentemente dos programas da família Bacon e de Glauber, BOOLE2 não realiza descoberta a partir de dados, mas a partir de uma representação abstrata de algum conhecimento científico [Ledesma 97].

Muitos sistemas de descoberta científica foram construídos para redescobrir conhecimentos já tidos como verdadeiros. Agora, a partir das bases lançadas pelos sistemas de simulação, acredita-se que um grande avanço pode ser conseguido no sentido de novas descobertas. Amostras disso já podem ser encontradas, como nos trabalhos de E. Mephu N'guifo [Mephu 93], que propôs um sistema de descoberta baseado em aprendizagem indutiva e em um método interativo de construção de teorias. Tal sistema tem sido utilizado na área de seqüenciamento genético.

2.5 Sistemas de construção interativa de teorias

A intervenção do usuário tem um papel importante no processo de descoberta científica feita por sistemas artificiais. Na verdade, esta nem sempre foi a mentalidade de pesquisadores e críticos, que julgavam a automatização completa do processo de descoberta como o principal objetivo da área. A idéia de construir sistemas que trabalhem em colaboração com o projetista aparece hoje como algo perfeitamente aceitável, especialmente se o objetivo for descobrir conhecimentos que sejam até então ignorados [Langley 98, Vades-Pérez 99].

A evolução das pesquisas em descoberta científica levou a sistemas projetados para a descoberta em áreas tais como Biologia, Química e Matemática. Devido a grande diversidade de sistemas, o trabalho de encontrar aspectos comuns a cada um deles tornou-se uma tarefa não trivial. Em [Langley 00], é proposto que uma atividade de descoberta (construção de sistemas de descoberta) seja composta por cinco estágios:

- (i) *Formação de taxonomias.* Leis/teorias formuladas devem ser relacionadas a conceitos e/ou categorias básicas. As melhores taxonomias vêm da Biologia, que agrupam entidades vivas em categorias e subcategorias hierarquicamente.
- (ii) *Descoberta de leis qualitativas,* que caracterizam o comportamento de um conjunto de entidades anteriormente identificadas e relacionam umas às outras, como por exemplo, ácidos tendem a reagir com os metais alcalinos para formarem sais.
- (iii) *A descoberta de leis quantitativas* estabelece relações matemáticas entre variáveis numéricas, como por exemplo, as massas relativas do ácido clorídrico e hidróxido

de sódio combinam-se para formar uma unidade de massa de cloreto de sódio e uma unidade de massa de água.

- (iv) *Criação de modelos estruturais*, que procuram/explicam regularidades entre entidades em observação, como os modelos atômicos e moleculares de Dalton e Avogadro.
- (v) *Criação de modelos de processo*, que explicam fenômenos em termos de mecanismos hipotéticos que envolvem mudanças sobre o tempo, como a teoria cinética dos gases.

Embora o termo *descoberta científica computacional* sugira um processo automatizado, uma inspeção mais detalhada revela que o desenvolvedor/usuário humano tem um papel importante em qualquer projeto bem-sucedido. Os passos no processo de descoberta em que o agente humano pode influenciar o comportamento do sistema são (figura 2.3) [Langley 00]:

- (i) *Formulação do problema*. O problema de descoberta pode ser formulado de tal forma que possa ser solucionado a partir de técnicas existentes. Deve-se, inicialmente, moedar a tarefa tais como aquelas que envolvem a formação de taxonomias, descoberta de leis qualitativas, detecção de relações numéricas, formação de modelos estruturais ou descrição de processos de construção.
- (ii) *Engenharia de representação*. É preciso estabelecer uma representação, ou seja, as características do domínio codificada em um formalismo. Deve-se estabelecer as variáveis e/ou predicados usados para descrever os dados/fenômenos a serem explicados, com a representação de saída usada por taxonomias, leis ou modelos.
- (iii) *Manipulação de dados*. Deve-se preparar os dados/fenômenos em que o sistema de descoberta vai operar, verificando fatores tais como a presença de ruídos e a falta de certos valores.

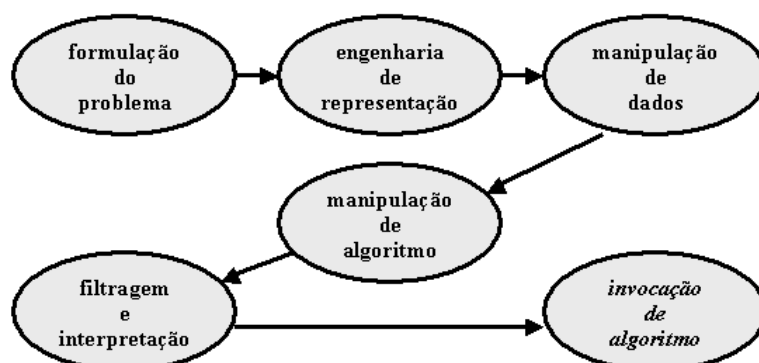


Figura 2.3: A importância do agente humana no processo de descoberta científica vista em seis passos.

- (iv) *Manipulação/invocação de algoritmo.* Deve-se realizar um esforço para modularizar o comportamento do algoritmo frente às entradas e ao agrupamento de parâmetros do sistema. Tal comportamento envolve fatores tais como tolerância a ruídos, critérios de parada e controle interativo da busca heurística para rejeitar maus candidatos ou atender aos bons.
- (v) *Filtragem e interpretação.* Deve-se, finalmente, transformar as saídas do sistema de descoberta em resultados que sejam significativos e compreensíveis pela comunidade científica.

Eis alguns exemplos de sistemas de descoberta científica desenvolvidos com base nessa abordagem:

- **Dendral.** Esse sistema foi desenvolvido para inferir a estrutura de componentes químicos em termos das conexões moleculares entre seus constituintes elementares, uma vez que para componentes complexos, pode haver centenas de milhões de possíveis estruturas. Dendral usa o espectro de massa para inferir subestruturas da molécula que poderia explicar os maiores picos nos dados (o sistema também pode aceitar entradas dos químicos), considera diferentes combinações destas subestruturas (mais os átomos residuais) que produzem a fórmula química conhecida (usando conhecimento da estabilidade química para gerar todos os grafos de estruturas químicas consistentes com estas restrições) e, por último, enfileira estes modelos estruturais candidatos em termos de suas habilidades para predizer os espectros observados usando conhecimento da espectrometria de massa para este propósito. Embora os trabalhos anteriores ao Dendral tenham enfatizado a automação do processo de modelagem estrutural, os desenvolvedores do sistema codificaram considerável conhecimento sobre estabilidade química em suas restrições de busca [**Feigenbaum 71**].
- **Autoclass.** Este programa foi projetado para formar taxonomias de um nível, agrupando objetos em classes ou *clusters* baseados em similaridades entre valores de atributo. Em um nível mais alto, Autoclass faz iterações através de diferentes números de clusters para determinar a melhor taxonomia, começando com um número de classes especificado pelo usuário e aumentando esta conta até produzir classes com probabilidades negligenciadas. A aplicação de Autoclass para dados infravermelhos produziu 77 classes estelares, que os desenvolvedores organizaram em 9 clusters de alto nível. A taxonomia difere significativamente daquela usada na Astronomia. Os

astrônomos colaboradores do projeto perceberam alguns resultados importantes – uma classe de estrelas de corpo negro com significante excesso de infravermelho, supostamente devido à poeira, e uma fraca “colisão” em 13 microns em algumas classes que estavam indetectáveis em espectros individuais [Cheeseman 88].

- **Graffiti.** Este sistema gera conjecturas em Teoria dos Grafos e outras áreas da matemática discreta. Graffiti assegura suas conjecturas como recentes mantendo um arquivo de hipóteses anteriores, e filtrando algumas conjecturas desinteressantes por notar que parecem ter sido implicadas, geralmente, pelas candidatas mais recentes. O sistema tem gerado centenas de novas conjecturas na Teoria dos Grafos, algumas das quais têm incitado matemáticos a se esforçarem para prová-las ou refutá-las, como por exemplo, a conjectura segundo a qual *a distância média de um grafo não é maior que seu número de independência* [Fajtlowicz 88].
- **Mechel.** Este sistema foi desenvolvido com o propósito de formular uma seqüência de passos, conhecida como *caminho da reação*, para uma dada reação química. O sistema aceita como entrada os reagentes e produtos para uma reação química, outras evidências experimentais e um considerável conhecimento acerca do domínio de Química Catalítica. Cabe ao usuário especificar interativamente as restrições a serem incorporadas durante a geração dos caminhos, dando a ele controle sobre o comportamento global do sistema [Valdés-Pérez 95].
- **RL.** Este sistema, baseado em indução de regras, foi aplicado a problemas de descoberta de leis qualitativas no estudo das causas de câncer por certas substâncias químicas (cancerígenas). O sistema foi executado para três bases de dados a fim que os resultados dos cancerígenos fossem analisados, incluindo 301, 108 e 1300 componentes químicos, respectivamente. Eles também testaram a habilidade das regras para classificar 24 novos componentes químicos cujo *status* era anteriormente, gerando também resultados positivos [Lee 98].

Como os exemplos acima mostram, o trabalho em descoberta científica não se focaliza somente em modelos históricos, mas também contribui para a geração de novos conhecimentos em uma vasta gama de disciplinas. Apesar de tais aplicações constituírem-se mais uma exceção do que uma regra, os sistemas de apoio à descoberta bem sucedidos provêem uma evidência de que estes métodos têm grande potencial para auxiliar o processo científico. A importância do agente humano em cada uma destas aplicações evidencia

seu papel imprescindível na atividade de descoberta.

2.6 Conclusão

O processo de descoberta científica é visto como uma criação da mente humana. Então, para algumas pessoas, incluindo os próprios cientistas, este parecia não ser um bom candidato para uma possível automação feita por computador. No entanto, nas últimas duas décadas, pesquisadores da Inteligência Artificial têm repetidamente questionado esta atitude e tentado desenvolver sistemas artificiais capazes de fazer descobertas em várias áreas da ciência: Matemática (como sistema AM [**Lenat 83a, Lenat 83b**]), Física (como sistema Bacon [**Langley 81**]), Química (como Dendral [**Feigenbaum 71**]), etc. Alguns pesquisadores têm tentado reproduzir descobertas históricas, enquanto outros têm despendido suas energias no sentido de projetar sistemas que encontrem conhecimentos até então ignorados. O estudo da descoberta científica computacional tem caminhado a passos largos.

Neste capítulo, foram mostrados alguns exemplos que ilustram a importância da intervenção de agentes humanos na atividade de descoberta realizada por um sistema artificial. A seguir é focalizado o SAID, um sistema inteligente concebido para ter a capacidade de construir seu próprio conhecimento através da interação com um agente humano e explicar suas decisões [**Nóbrega 98**]. É apresentada também o ϕ -calculus, uma abordagem teórica e formal de gerência do raciocínio no contexto das ciências empíricas [**Sallantin 00, Nóbrega 00**].

Capítulo 3

Abordagens teóricas para a construção interativa de teorias

3.1 Introdução

Apesar das tentativas de se definir aquilo que se chama *inteligência*, não se tem ainda um consenso nas comunidades científicas concernidas. Existe, no entanto, seres que possuem comportamentos que, por convenção, são tidos como inteligentes: os *agentes inteligentes*.

A busca pela conceituação e pela concepção de sistemas (ou agentes) artificiais capazes de exibir um comportamento tido como inteligente deve, portanto, considerar diversos princípios que subentendem esse comportamento. Entre essas noções, destaca-se a de *racionalidade* [Newell 62], citada no capítulo 2. Mais específica que inteligência, a noção de racionalidade está ligada ao tratamento de uma classe de problemas bem delimitada. Neste trabalho, considera-se como racionais os agentes (humanos ou artificiais) capazes de produzir e controlar seus próprios conhecimentos sobre um certo domínio. Para ter essas características, um tal sistema deve dispor de funcionalidades para classes de tarefas como decidir, classificar, diagnosticar, predizer, simular, contradizer, conceber ou planificar [Ferneda 92b].

Mesmo hoje, não existe ainda uma definição única para a noção de *agente*. Porém, boa parte dos pesquisadores associa tal conceito à idéia de uma entidade (software ou hardware) que, imersa em um ambiente, percebe e age sobre ele [Russell 95]. Um agente também é capaz de modificar sua representação interna a partir desta interação com o ambiente e com outros agentes [Ferber 99]. O conceito de agente vem sendo utilizado para várias finalidades, entre elas aplicações em comércio eletrônico, em gerenciamento de in-

formações e correio eletrônico.

Entre os diversos trabalhos realizados e em andamento sobre o tema de agentes racionais, no projeto SAID [Ferneda 92a] busca-se a construção de um *agente racional* visto como um sistema que aparenta ao usuário ser capaz de raciocínio como resultado da manipulação de interações, de crenças e de conhecimento incompleto, impreciso e errôneo. Este agente combina Aprendizagem de Máquina e Aquisição de Conhecimento para gerar seu próprio conhecimento, controlar a evolução desse conhecimento e explicar suas decisões na interação com um especialista do domínio. Esforços foram realizados no sentido de uma especificação formal [Nóbrega 98] e de uma implementação [Carvalho 99, Gheyi 00].

Por outro lado, ϕ -calculus é uma ferramenta teórica para concepção interativa de uma maneira geral. A idéia é que um especialista do domínio disponha de um sistema (um agente artificial) como aprendiz e colaborador, dada sua habilidade para processamento, no sentido de ajudá-lo a raciocinar dentro de seu domínio de trabalho e, talvez, contribuir para refinar a representação mental de seu instrutor [Nóbrega 00, 01a, 01b, 02a, 02b].

Neste capítulo, será apresentado o projeto SAID e sua evolução para o projeto ϕ -calculus.

3.2 O projeto SAID

Um agente pode ser visto como uma entidade que, entre outras coisas, percebe e age. Agir *racionalmente* significa agir de maneira a atingir objetivos de acordo com um conjunto de crenças [Nóbrega 98]. Não existe uma única definição de agente racional aceita universalmente. Para alguns estudiosos, agente racional é:

“... um agente que busca efetuar as ações as mais pertinentes possíveis para atingir seus objetivos em função do conhecimento que ele possui de suas ações e do ambiente”. [Müller 87]

“... um sistema inteligente autônomo, capaz de raciocínios do senso comum como os que se exercem na vida cotidiana, e que fazem intervir intenções, crenças e conhecimentos incompletos e errôneos”. [Ferneda 92a]

“... um sistema que dispõe de métodos para produzir e controlar seus próprios conhecimentos na interação de um especialista do domínio”. [Sallantin 97]

“... aquele que faz a coisa certa”. [Russell 95].

Pode-se dizer então que um agente racional deve ser capaz de construir e fazer evoluir seus conhecimentos em certo domínio, ser autônomo (no sentido de que suas ações são determinadas pelo seu conhecimento passível de evolução) e procurar agir de forma a atingir seus objetivos.

O agente racional SAID (Somente Abduzir, Induzir e Deduzir) é um sistema inteligente com a capacidade de construir seu próprio conhecimento através da interação com um agente humano. Esse sistema é capaz de fornecer explicações sobre suas decisões e pode ser visto como um sistema de aquisição de conhecimento organizado em três níveis hierárquicos:

- Um *sistema de representação de conhecimento* que provê os meios necessários à construção de raciocínios sobre conhecimentos incompletos e passíveis de erros (as Teorias Semi-Empíricas) [Sallantin 91a, Sallantin 91b],
- Uma *metodologia* de integração de diversos mecanismos de aprendizagem para a geração do conhecimento: abdução, indução e dedução (figura 3.1) [Villareal 89, Liquière 90, Ferneda 92a, Carvalho 99];
- Um *ambiente multi-agente* que provê os elementos necessários para o controle por diálogo entre os diversos agentes envolvidos na geração do conhecimento (MOSCA) [Reitz 92, Gheyi 00].

O agente racional SAID é um sistema de apoio à descoberta científica.

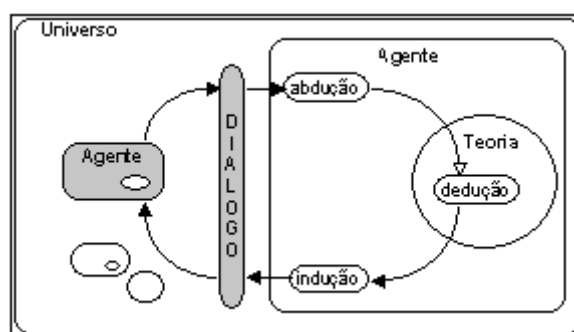


Figura 3.1: Mecanismos do agente racional SAID

3.2.1 Teorias Semi-Empíricas

TSE é uma forma de representação do conhecimento que expressa uma conjectura através de relacionamentos e restrições sobre relacionamentos entre sentenças de uma linguagem. O modelo da descoberta científica adotado pelas TSE objetiva o estudo da prova

(ou refutação) de uma conjectura para determinar o que intervém no exame da validade (dado o seu poder preditivo) da prova (ou refutação) e ainda o que intervém no exame da relevância (dado o seu poder explanatório) da prova (ou refutação). Na TSE os conhecimentos são classificados como (figura 3.2):

- *Dados*, que representam o conhecimento de uma agente racional. O conhecimento é capturado em uma das três formas seguintes: *fatos*, *hipóteses* e *heurísticas*.
- *Mecanismos* para a geração de conhecimento (por *abdução*), para a organização desse conhecimento (por *indução*) e para a propagação de restrições sobre ele (por *dedução*).
- *Métodos* relacionados às interações com um agente externo que desempenha o papel de criticar ou de propor uma sentença a ser provada.

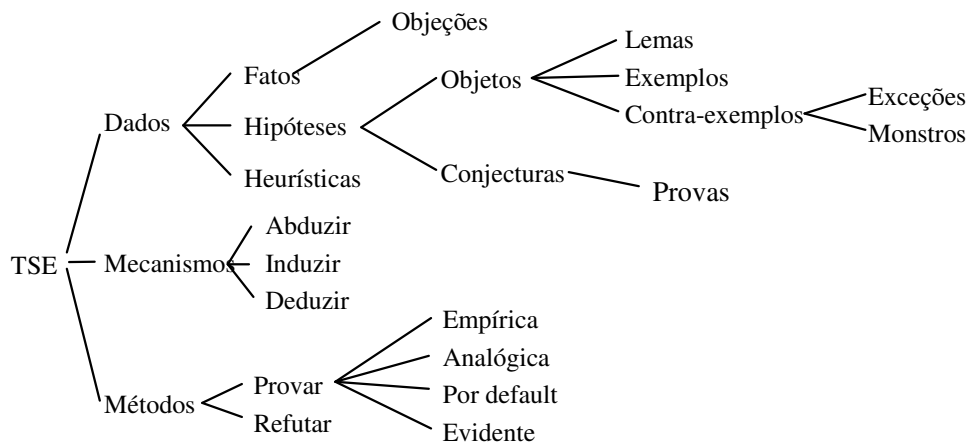


Figura 3.2: Termos que intervêm na formalização e na evolução do conhecimento pelas TSE.

As definições para os termos usados na taxonomia da TSE são:

- *Objetos*: enunciados de uma restrição particular que satisfazem um conjunto de fatos.
- *Conjecturas*: enunciados de uma restrição particular que satisfazem um conjunto de objetos.
- *Lemas*: enunciados conjuntivos que expressam fatos concomitantes; ou disjuntivos que expressam fatos excludentes.
- *Prova*: uma decomposição de conjecturas em um conjunto de lemas;
- *Refutações*: enunciados de contra-exemplos.
- *Exemplos*: objetos que satisfazem a conjectura e os lemas da prova.
- *Contra-exemplos*: objetos que não satisfazem a conjectura ou a prova.
- *Objecções*: enunciados de diferenças entre exemplos que satisfazem a conjectura e ob-

jetos já criticados pela conjectura.

- *Exceções*: objeções aceitas para uma conjectura, mas não para sua prova.
- *Monstros*: objeções aceitas para uma prova, mas não para a conjectura.
- *Provas evidentes*: provas que usam argumentos construídos com dados iniciais.
- *Provas por default*: provas dadas como argumentos alternativos na ausência de uma prova evidente ou construída pelo aprendiz.
- *Provas empíricas*: provas que utilizam lemas obtidos por raciocínio empírico.
- *Provas analógicas*: provas que utilizam lemas obtidos por raciocínio analógico.

Um conhecimento nas TSE é avaliado segundo os critérios de *validade* e *pertinência*. A validade é uma avaliação dos fatos produzidos pelas hipóteses. A validade dos fatos pode ser diretamente mensurada, por exemplo, através da determinação da frequência da verificação de um fato por uma classe de objetos. A pertinência é uma avaliação da maneira pela qual as hipóteses produzem os fatos, refletindo o número de resultados provados, aceitos pelo pesquisador. As variações das medidas associadas aos critérios de validade e pertinência, em função da teoria, são chamadas, respectivamente, de *interesse* e *simplicidade*.

3.2.2 Esquema Mental

A linguagem formada pelo conjunto de sentenças é usada tanto para capturar como para comunicar conhecimento. Visando a definição de objetos como interpretações de um conjunto de sentenças e de conjecturas como interpretações dos conjuntos de objetos, apresenta-se a noção de esquema mental. Cada agente racional é composto de uma TSE e um esquema mental. Um esquema mental é uma tripla (L, C, Δ) , onde L é um conjunto de sentenças, C é um conjunto de valores de crenças e $\Delta: L \times L \rightarrow C$ é uma função de crenças definida como: se s_1 e s_2 são duas sentenças e $C = \Delta(s_1, s_2)$, então a sentença s_1 explica (com crença C) a sentença s_2 no esquema mental.

3.2.3 Um protocolo para aprendizagem

Um ambiente mínimo de aprendizagem é composto de um *aprendiz* que se comunica com um *oráculo*. O protocolo que rege o diálogo, do ponto de vista de resolução de problemas, pode ser resumido como: o oráculo envia problemas resolvidos ao aprendiz e este último, ao receber tais problemas, reúne-os em uma *amostra*. Dada uma amostra, o aprendiz procura, em um *espaço de hipóteses*, a melhor *hipótese* em *adequação* à amostra.

Hipótese, neste contexto, significa um procedimento de resolução de problema. O conceito de adequação é modelado como um critério a ser satisfeito, um critério de aprendizagem, que estabelece o que significa uma melhor hipótese em uma dada amostra. A hipótese selecionada é chamada *hipótese aprendida* (ou *conjectura*, segundo a TSE). Um espaço de hipóteses representa todo o conhecimento que um aprendiz pode obter e, em um dado instante, o seu conhecimento é um ponto nesse espaço de hipóteses. Os dados de um problema de aprendizagem se resumem em:

- Uma amostra de problemas resolvidos,
- Um espaço de hipóteses,
- Um critério de aprendizagem (adequação hipótese-amostra) e
- Uma estratégia de percurso do espaço de hipóteses.

Dois tipos de ruídos na amostra irão se somar à complexidade inerente do processo de busca de uma hipótese:

- O par $\langle \text{problema}, \text{solução} \rangle$ pode estar descrito incorretamente,
- A linguagem adotada para descrever estes pares pode não ser suficiente para discriminar enunciados de problemas distintos e conseqüentemente com soluções diferentes.

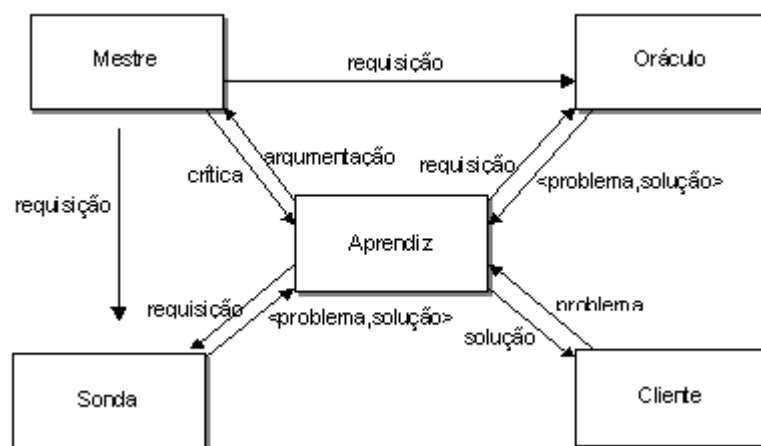


Figura 3.3: Protocolo de aprendizagem MOSCA do ponto de vista de resolução de problemas.

No protocolo MOSCA, destacam-se cinco papéis (figura 3.3):

- *Aprendiz*, que constrói uma hipótese a partir da amostra de exemplos e contra-exemplos;

- *Oráculo*, que produz soluções irrefutáveis de problemas, na forma de pares $\langle \text{problema}, \text{solução} \rangle$
- *Cliente*, que submete problemas ao aprendiz e espera soluções;
- *Sonda*, que produz soluções refutáveis na forma de pares $\langle \text{problema}, \text{solução} \rangle$, obrigando o aprendiz a produzir argumentações;
- *Mestre*, que analisa as argumentações do aprendiz e as critica.

3.2.4 Sistema de crenças

O sistema de crenças considerado baseia-se na noção de bi-reticulado de Ginsberg [Ginsberg 90]. Uma crença é um elemento do conjunto C , munido de duas relações de ordenação: \leq_k e \leq_t , que interpretam-se, respectivamente, como *menos conhecido que* e *menos verdade que*. Pela relação \leq_k , C é um reticulado com *silêncio* como seu mínimo e com \perp (contradição) como seu máximo. Pela relação \leq_t , C é um reticulado com *verdadeiro* como máximo e *falso* como mínimo. A crença \perp é tal que *verdadeiro* $\leq_k \perp$ e *falso* $\leq_t \perp$, representando uma contradição.

O conjunto C de crenças considerado é

$$C = C_M \cup C_O \cup C_S \cup C_C \cup C_A,$$

onde:

$$C_M = \{\text{objetado}, \text{não-objetado}, \text{silêncio}\},$$

$$C_O = \{\text{verdade-O}, \text{falso-O}\},$$

$$C_S = \{\text{verdade-S}, \text{falso-S}\},$$

$$C_C = \{\}$$
 e

$$C_A = \{\text{aceito}, \text{contestado}, \text{silêncio}, \text{contradição}\}.$$

A ordem entre essas crenças é dada como na figura 3.4, que pode ser interpretada segundo:

(i) a ordem das crenças de conhecimento

$$\text{silêncio} \leq_k \text{verdade-S} \leq_k \text{aceito} \leq_k \text{não-objetado} \leq_k \text{verdade-O} \leq_k \perp$$

e

$$\text{silêncio} \leq_k \text{falso-S} \leq_k \text{contestado} \leq_k \text{objetado} \leq_k \text{falso-O} \leq_k \perp$$

(ii) a ordem das crenças de verdade

falso-O \leq_t objetado \leq_t contestado \leq_t falso-S \leq_t silêncio \leq_t
 verdade-S \leq_t aceito \leq_t não-objetado \leq_t verdade-O

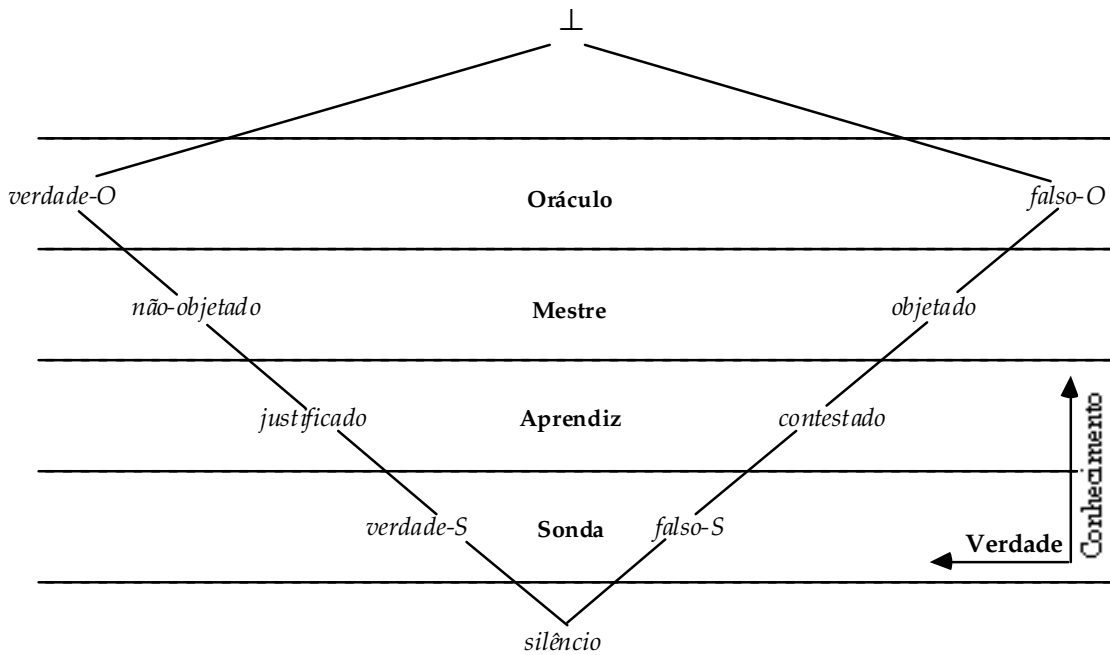


Figura 3.4: Sistema de crenças considerado.

3.2.5 Princípios para a geração e evolução do conhecimento

No protocolo MOSCA, a forma dos problemas que o aprendiz pode resolver se restringe à identificação de conceitos. Os dados disponíveis de um problema típico são um objeto e um conceito e, sua solução, uma crença indicando se o objeto é ou não um exemplo do conceito. O aprendiz possui mecanismos para a construção de uma conjectura, que são formadas a partir de combinações lógicas de componentes elementares (regularidades).

Princípios para a construção de uma conjectura

O objetivo do aprendiz é construir uma conjectura sobre um conceito. Assume-se que a única informação que o aprendiz recebe é uma amostra composta de exemplos e contra-exemplos deste conceito, oferecidos pelo oráculo, e constrói, observando regularidades na amostra, regras na forma:

“Se o objeto satisfaz a regularidade R, então sentença S tem crença C”.

Princípios para a aplicação de uma conjectura

Quando a sonda ou o oráculo apresenta um problema resolvido ao aprendiz, este

determina uma crença da pertinência de um objeto ao conceito em dois passos:

- (*passo 1*) Aplicação de um conjunto de regras que formam a conjectura, resultando em um conjunto de crenças, uma para cada regra aplicada e,
- (*passo 2*) Agregação dos valores de crença em uma única crença, que será associada ao problema apresentado (um par <*objeto, conceito*>).

Essa agregação pode ser realizada de várias formas:

- *A abordagem clássica*: se um conjunto de valores de crença é {verdadeiro, falso}, então a agregação pode ser feita por conjunção, como por disjunção,
- *A abordagem da lógica majoritária*: dois limites permitem regular a natureza mais conjuntiva ou mais disjuntiva da agregação das regras aprendidas.

O primeiro limite indica o número mínimo de regras que devem ser verdadeiras para que a agregação também o seja. O segundo limite indica o número mínimo de regras que devem ser falsas para que a agregação também seja falsa. Como esses limites não são necessariamente complementares, para alguns casos o resultado não será verdadeiro nem falso, será *silêncio*.

Princípios para a construção de uma argumentação

Uma revisão do conhecimento aprendido é empreendida em dois casos:

- Ou um novo exemplo/contra-exemplo é proposto pelo oráculo, o qual vem enriquecer a amostra do aprendiz, o que questiona as regularidades observadas (aparição de novas regularidades, desaparecimento de antigas) e, portanto, da conjectura aprendida.
- Ou a argumentação produzida pelo aprendiz é criticada, o que, indiretamente, questiona essa mesma conjectura.

O primeiro caso diz respeito ao estudo da natureza incremental dos mecanismos de detecção de regularidades na amostra. Esse problema clássico da aprendizagem é grande consumidor de recursos de cálculo e portanto é preferível que não seja utilizado como forma de revisão. Adotou-se o caso da argumentação.

Princípios de exploração de uma crítica

Quando o aprendiz é criticado sobre uma argumentação, é possível a ele classificar as regras aprendidas das seguintes formas:

- Certas regras não foram ainda implicadas na argumentação;
- Outras foram utilizadas para se construir argumentações, as quais foram criticadas ou não.

A gestão da argumentação gera um certo número de problemas que são mostrados a seguir:

- A revisão de uma hipótese aprendida deve questionar as informações sobre as críticas recebidas na função de argumentação;
- A exaustão das argumentações possíveis, quando o aprendiz é sempre criticado, deve conduzir à revisão de sua conjectura, mesmo quando ela está em perfeita adequação com os dados da amostra;

Uma argumentação criticada questiona mais o modo de argumentação do que os argumentos que a compõem.

3.2.6 Discussão

Alguns trabalhos em descoberta científica têm sido direcionados para a concepção e o desenvolvimento de um sistema interativo de apoio à descoberta baseado na noção de agentes racionais [Ferneda 92b]. O agente racional SAID, proposto por Jean Sallantin e sua equipe [Sallantin 91a, Sallantin 91b], é um sistema inteligente com a capacidade de construir seu próprio conhecimento através da interação com um agente humano, como foi visto. Neste sentido, Germana Menezes da Nóbrega propôs uma especificação formal de um tal sistema [Nóbrega 98]. Isso representou um ponto de partida para uma implementação, realizada no âmbito de projetos de Iniciação Científica de alunos do Curso de Bacharelado em Ciência da Computação da UFPB [Gueyi 00], que proporcionará, por um lado, o experimento de algoritmos de aprendizagem a partir de exemplos em vários domínios e, por outro lado, uma ferramenta de apoio à geração e validação de teorias científicas.

Foi também desenvolvido um método de aprendizagem de máquina para a busca de similaridades em objetos estruturados [Martins 98, Carvalho 99], responsável pela pri-

meira fase na construção de uma teoria sobre um domínio tal como a geometria euclidiana plana. A geração completa de uma conjectura sobre um conceito está agora em fase final de desenvolvimento [Carvalho 01]. Neste sentido, este trabalho vai ao encontro dos atuais esforços realizados no âmbito de disciplinas como Aquisição de Conhecimento e Aprendizagem de Máquina.

A geometria foi escolhida como campo de aplicação pois é evidente o acúmulo de reflexões sobre o tipo de raciocínio praticado nesse domínio, seja sob a formulação de enunciados de uma conjectura, seja sobre a forma de se dar a uma prova, sobre a influência das figuras, das construções e das abstrações. Além disso, os desenvolvimentos do cálculo formal em geometria permitem bem situar a nossa empreitada face a uma abordagem algorítmica do raciocínio. Em [Ferneda 92b] já foi examinada a pertinência de uma conceitualização do raciocínio para um agente racional pelos especialistas do domínio da Geometria, em particular aqueles interessados em seus aspectos pedagógicos. Apoiou-se na especificidade gráfica da Geometria para uma ilustração detalhada do conjunto de procedimentos. Ilustrou-se efetivamente, através de desenhos, aquilo que o agente racional aprenderia por abdução, as hipóteses que ele induziria, assim como as conjecturas que ele proporia. Falta, porém, uma experimentação efetiva, haja vista que apenas parte dessa abordagem pode ser testada na época.

Uma proposta de evolução da concepção de agentes inteligentes foi ainda proposta por Mário Ernesto de Souza e Silva em [Silva 98], que estudou a evolução da concepção de um agente racional para um sistema geral de aprendizagem por experiência baseada na teoria Semiótica de Charles Sanders Peirce. Foi realizada uma análise conceitual da concepção do SAID numa perspectiva da Semiótica e dos resultados alcançados em trabalhos anteriores sobre a concepção de sistemas inteligentes nela baseados.

3.3 O projeto ϕ -calculus

Trabalhos recentes realizados no LIRMM¹, sob a direção de Jean Sallantin, sobre a construção interativa de teorias abordam as noções de *racionalidade* e *adequação* para investigar a atividade de concepção quando um agente humano utiliza um agente artificial para explicitar um conhecimento coerente e livre de ambigüidades [Nóbrega 00, 02a]. Tais trabalhos apóiam-se em três vertentes complementares – *conceitual*, *formal* e *experimental*

¹ LIRMM = Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier, França.

– para estabelecer e validar uma metodologia para a concepção interativa.

Na *vertente conceitual*, a meta é investigar como a noção de racionalidade pode ser explorada para possibilitar coerência e ausência de ambigüidade no conhecimento durante seu processo de explicitação. A *vertente formal* vai ao encontro da verificação das propriedades que um tal processo demandaria em termos de linguagens de representação que facilitem a intervenção do agente humano, bem como a verificação por este da adequação do conhecimento explicitado. A originalidade dessa abordagem, no contexto da IA, repousa no fato de que nenhum formalismo lógico conhecido apreende diretamente a interação no processo de explicitação de conhecimento passível de evolução [Sallantin 00]. As *vertentes experimentais* ora realizadas vão ao encontro de domínios tais como Informática na Educação e Comércio Eletrônico. Nesse contexto, ϕ -calculus é proposto como uma alternativa metodológica para dar suporte ao processo de construção de teorias sob supervisão humana.

A noção de racionalidade proposta por Newell [Newell 62] consiste em usar o melhor meio possível para limitar os espaços de busca a proporções manipuláveis, quando se está em um processo de descoberta de uma teoria. A noção de adequação enfatiza o julgamento humano com relação às representações do mundo, de acordo com seu ponto de vista. A idéia é unificar, em um mesmo formalismo, estas duas abordagens guiadas pelo cálculo e pela adequação segundo a interpretação do usuário. A teoria desenvolvida fornece bases formais para a construção de teorias.

3.3.1 Motivação inicial: a origem do método

Desde 1994, ϕ -calculus tem sido experimentado em colaboração com a companhia de advogados **FIDAL-KPMG**, composta de 120 advogados na França. A atividade diária destes profissionais consiste em compreender, provar e comparar contratos. Como as leis, normas e eventos mudam continuamente, é necessário que os contratos tenham que ser modificados de acordo – qualquer novo contrato requer necessariamente um compromisso entre a liberdade de negócio entre as partes envolvidas, e a segurança deste negócio com respeito a eventos não previstos durante o tempo de construção do mesmo. No sentido de auxiliar os advogados em suas atividades, a companhia tem identificado classes de contratos, e para cada destas classes tem criado um *template*. É a construção destes *templates* que é assistido pela metodologia ϕ -calculus e sua ferramenta computacional, chamada de

fid@act. Esta ferramenta assiste o processo interativo de construção de um *template* para cada classe de contratos legais por explicar inadequações, no sentido de gerar um texto – o *template* é um padrão de classe de instancias de contrato – julgado coerente e seguro, como descrito acima.

A companhia delega o projeto do *template* a um advogado sênior, experiente, e dois ou três advogados juniores, familiarizados com a ferramenta fid@act. O trabalho dos juniores consiste em: (primeira fase) avaliar um conjunto pré-existente de contratos na classe; (segunda fase) extrair termos tais como as noções de *franchise*, duração, obrigação e sanção, codificando as prescrições do sênior na forma de restrições lógicas entre os termos, e (terceira fase) formular um *template* correto sintaticamente, respeitando as prescrições do sênior. Sua atitude ao fazer isto é desafiar o sênior, por gerar um *template* que respeita as restrições sintáticas, mas ao mesmo tempo semanticamente absurdo. O sênior reage modificando sua própria teoria, enquanto os juniores revisam o conjunto de termos e prescrições.

O processo dura em média 10 encontros de 2 horas separados por uma semana. Durante a primeira fase as revisões focalizam-se nos termos – entre 800 e 1800, geralmente próximo de 1200 – e as restrições associadas a presença e/ou ausência em uma fórmula. Durante a segunda fase o sênior reage ao contrato absurdo dos juniores fixando o contexto de uso de uma fórmula. Durante a terceira fase o sênior explica aos juniores por que o contrato construído é correto (sintaticamente) e absurdo (semanticamente), sugerindo que sejam introduzidas novas prescrições e fixando o uso do *template* - o objetivo é eliminar os *erros* de denominação, vindo dos termos, as *ambigüidades* de conotação, vindas das fórmulas, e os *paradoxos*, vindos dos exemplos.

Para a construção de um único *template* são necessários, em média, três meses de trabalho, mas, assim que fica disponível, um advogado treinado, familiarizado com a ferramenta fid@act, pode conceber e escrever um novo contrato em quinze minutos. 400 advogados estão usando o *template* gerado a partir da aplicação da metodologia descrita. Uma patente tem sido requerida pela companhia [FIDAL 00].

3.3.2 Fundamentos teóricos

φ -calculus é uma abordagem teórica e formal tentativa de gerenciar raciocínio assíncrono interativo no contexto das ciências empíricas [Sallantin 00]. A idéia do raciocínio

assíncrono interativo é a seguinte: um instrutor (um ser humano especialista do domínio), tendo um certo conhecimento sobre um determinado assunto, está interessado em rever e fazer evoluir este conhecimento. Como colaborador, o instrutor dispõe de um agente artificial, sistema que aprenderá (e raciocinará sobre) o conhecimento que o instrutor detém. É importante verificar como o conhecimento será representado e evoluído dentro do agente artificial.

Após a fase de *ensino*, duas outras se iniciam: a *avaliação do conhecimento* (do agente artificial pelo especialista do domínio) e a *atividade de correção* (quando acontece o refinamento do conhecimento aprendido). Este processo de colaboração/aprendizagem entre o especialista do domínio e o agente artificial é assíncrono e interativo.

Entende-se por *conhecimento* todas as formas de saber do homem: objetos que formam o mundo real, fatos, conceitos mais vastos (que correspondem a grupamentos ou generalizações de objetos de base), relações entre conceitos, heurísticas e procedimentos de raciocínio – o conhecimento pode ser um problema específico, geral a um domínio, exato, impreciso ou incompleto. A isto se adiciona o *meta-conhecimento*, relativo à confiança em seu conhecimento e à visão que se tem de seu próprio raciocínio, ou seja, “conhecimento sobre conhecimento”. O problema da *representação de conhecimento* é o de sua transição sob uma forma simbólica que possa ser explorada por um sistema de raciocínio. O modo de representação associa assim dois aspectos imbricados, ou mesmo confundidos: (a) a estrutura de dados para representar a informação e (b) o método de exploração (ou de raciocínio) associado [Haton 90]. O que diferencia uma representação de conhecimento de um sistema "ad hoc" de manipulação de estruturas de dados é o fato da representação de conhecimento ter um significado externo, isto é, ser *sobre* alguma coisa [Bittencourt 96].

A ferramenta computacional interativa é baseada na categorização de *atitudes mentais matemáticas*, que ajudam um agente a racionalizar a construção de conhecimento. Uma semântica consiste em dar um conjunto de objetos matemáticos com a finalidade de compreender, provar e comparar propriedades computáveis do mundo. Categorização é usada, neste trabalho, com dois sentidos: (i) categorização natural, relacionada à faculdade de classificar os objetos do mundo e (ii) atitudes mentais matemáticas, usadas para realizar a categorização natural. O problema matemático é encontrar uma ferramenta para processar a formalização.

A metodologia do φ -calculus é constituída por três processos principais: Expressão, Formalização e Computação.

3.3.2.1 Expressão

A *Expressão* é o processo para desenvolver asserções, que são usadas para predizer ou explicar fenômenos. Segundo [Vuillemin 84], *asserção* comunica a alguém tanto o conhecimento como a experiência do assunto, e “fala” quais suas possíveis regularidades. Uma crença em uma asserção é uma adequação entre uma sentença e as proposições, ligadas à sentença e consistentes em um dado modelo. O termo *proposição* recorre a idéia de asserção formulada segundo uma certa sintaxe e susceptível de ser avaliado como “verdadeiro” ou “falso” por um universo dado [Haton 90].

SENTENÇA \leftarrow ADEQUAÇÃO \rightarrow PROPOSIÇÃO

Um agente aprendiz acredita em uma sentença se, em seu modelo, a resposta para a sentença é afirmativa; ao contrário, em caso de inadequação, a pergunta feita é POR QUÊ?

Atitudes mentais matemáticas

Um *fato* é uma asserção expressando algo que existe. As proposições de um modelo associadas à categoria de fato são chamadas de *leis*. Uma lei expressa algo universal que pode restringir um fato. Em Física, E (energia), M (massa) e c (luz) são fatos, enquanto $E = mc^2$ é uma lei. Estas atitudes mentais matemáticas referem-se ao conhecimento do domínio, ou discernimento de mundo, por parte do especialista, ou instrutor.

FATO \leftarrow ADEQUAÇÃO \rightarrow LEI

Uma *hipótese* é uma asserção expressando uma possibilidade – relação entre fatos supostamente pertinentes. A proposição do modelo associada à hipótese é chamada de *postulado*, que é definido como um conjunto de regras que sustentam, ou seja, dão suporte à hipótese.

HIPÓTESE \leftarrow ADEQUAÇÃO \rightarrow POSTULADO

Um *exemplo* é uma asserção associada à crença individual do especialista do domínio - os exemplos representam instâncias (positivas ou negativas) do conceito que ele deseja obter. O grande objetivo do especialista do domínio é construir um conceito que sinteti-

ze o conhecimento que ele deseja transmitir a seu agente, testando várias hipóteses e escolhendo aquela que o representará. Portanto, as proposições do modelo associadas aos exemplos constituem o *conceito*.

EXEMPLO ← ADEQUAÇÃO → CONCEITO

O processo de expressão é restringido por combinações internas: fatos são combinados para formular hipóteses (a hipótese deve estar em correspondência com os fatos que existem espontaneamente na natureza e que têm algum significado para o especialista), que são combinadas para formular exemplos (que são construídas a partir dos exemplos), e *exemplos* são combinados para formular *hipóteses* (os exemplos são uma representação em extensão da hipótese).

FATO → HIPÓTESE ↔ EXEMPLO

Uma inadequação provoca uma revisão do modelo – inadequação é usada aqui como sinônimo de violação entre uma sentença e sua proposição. Quando existe uma inadequação entre lei e fato, diz-se que houve um *desentendimento*. Uma *ambigüidade* é uma inadequação entre uma hipótese e um postulado. Já quando o problema é entre exemplo e conceito, houve um *paradoxo*.

3.3.2.2 Formalização

A *Formalização* é um processo indutivo interativo usado para desenvolver formalismos matemáticos em uma linguagem formal. O formalismo é usado para descrever modelos adequados. Neste processo, o instrutor tem acesso direto ao conhecimento aprendido/gerado pelo agente racional, no sentido de refina-lo.

Atitudes mentais matemáticas

Uma *opinião* expressa um julgamento acerca de uma crença falsa, verdadeira ou desconhecida. Este julgamento é feito sobre a universalidade dos *princípios*. Uma outra maneira do especialista do domínio obter o conceito (sem a necessidade de possuir uma amostra de exemplos, positivos ou negativos, e sem passar pela aprendizagem automática) é emitir uma opinião acerca do que ele pensa sobre o domínio. Os princípios são regras que dão suporte à opinião, ou seja, são suas partes constituintes.

Uma *conjectura* julga a generalidade do *lema*. A *objeção*, uma *prova*. Lemas são

regras obtidas a partir de uma amostra de exemplos. Conjectura é um conjunto de lemas – uma das conjecturas será a hipótese escolhida como aquela a ser testada. Objeção é uma discordância do especialista com relação ao conhecimento armazenado pelo agente, sugerindo modificações na hipótese que está sendo testada. Os termos que sustentam esta objeção são chamados de provas.

OPINIÃO ← CONSISTÊNCIA → PRINCÍPIO
CONJECTURA ← CONSISTÊNCIA → LEMA
OBJEÇÃO ← CONSISTÊNCIA → PROVA

Consistência é usada neste trabalho como sinônimo de estabilidade entre uma sentença e sua proposição, ou seja, a proposição é considerada como o elemento de sustentação de sua sentença.

O processo de formalização é restringido pela semântica: lemas são combinadas para formular provas, que por sua vez são combinados para formular princípios - princípios devem estar em correspondência com as provas. Estas combinações devem-se ao fato de que tanto os princípios, lemas e provas, mesmo sendo obtidos em momentos distintos da interação especialista–agente aprendiz, servirem de base para a origem dos postulados, os elementos de sustentação para a hipótese.

LEMA → PROVA ↔ PRINCÍPIO

Da mesma forma que:

LEI → POSTULADO ↔ CONCEITO

Representam formalmente o conhecimento detido pelo especialista do domínio e transmitido a seu agente aprendiz.

As conjecturas são combinadas para formular objeções, que são combinadas para formular opiniões. As opiniões devem está em correspondência com as objeções. Estas combinações devem-se ao fato de que tanto a opinião, conjectura e objeção, mesmo originadas em momentos distintos da interação especialista–agente aprendiz, servirem de base para a hipótese.

CONJECTURA → OBJEÇÃO ↔ OPINIÃO

3.3.2.3 Computação

Para a construção do modelo de adequação, usam-se mecanismos de indução e abdução. As adequações a serem verificadas são:

$$\text{FATO} \rightarrow \text{LEMA} \leftrightarrow \text{CONJECTURA} \rightarrow \text{LEI} \leftrightarrow \text{FATO} \quad (1)$$

$$\text{HIPÓTESE} \rightarrow \text{PROVA} \leftrightarrow \text{OBJEÇÃO} \rightarrow \text{POSTULADO} \leftrightarrow \text{HIPÓTESE} \quad (2)$$

$$\text{HIPÓTESE} \rightarrow \text{PRINCÍPIO} \leftrightarrow \text{OPINIÃO} \rightarrow \text{POSTULADO} \leftrightarrow \text{HIPÓTESE} \quad (2)$$

Por fim,

$$\text{EXEMPLO} \rightarrow \text{LEMA} \leftrightarrow \text{CONJECTURA} \rightarrow \text{CONCEITO} \leftrightarrow \text{EXEMPLO} \quad (3)$$

O ciclo (1) constrói uma *adequação ontológica*, o (2) uma *epistêmica*, e o (3) uma *heurística*. A *busca por adequação* procura aproximar ao máximo possível o conhecimento armazenado pelo sistema/agente aprendiz com aquele pensado pelo especialista do domínio. Cabe a este avaliar/decidir quando a adequação é obtida.

3.3.2.4 Busca por adequação

Para entender o processo de busca por adequação, considere a situação seguinte (ilustrada na figura 3.5): um supervisor tem uma *formulação* mental, uma *conceituação*, de um conhecimento no seu domínio para ser convertida de alguma maneira em uma *formalização* matemática, uma *ontologia*, passível de manipulação por uma máquina (seta 1). Considere ainda que tal formalização deva ser capaz de suportar *figurações* realizadas sobre a mesma máquina (seta 2) e estas, por sua vez, deveriam ser aceitas pelo supervisor como verificando sua formulação inicial (seta 4). Assim, a adequação é atingida quando uma figuração sobre uma formalização (seta 3) de uma formulação é aceita pelo supervisor como sendo uma figuração de sua formulação inicial. O conhecimento sobre o mundo não precisa, necessariamente, obedecer a nenhuma propriedade matemática global, como coerência ou completude. Sua adequação deve ser medida pela sua utilidade na solução de problemas práticos [Bittencourt 96].

Conceituação representa um assunto (de discussão) interessante para alguém (ou grupo), enquanto ontologia refere-se a estrutura julgada por uma pessoa (ou grupo) como apropriada para descrever (mesmo parcialmente) uma conceituação [Nóbrega 00]. A partir da noção de busca por adequação, a conceituação pode ser caracterizada como um processo interativo e construtivo. Durante este processo, significados são atribuídos às relações

na ontologia pelo especialista do domínio, enquanto uma representação em intenção para as relações é gradualmente construída.

Na prática, a adequação não é atingida de imediato - um número de iterações geralmente grande é necessário (podendo até ser infinito). Uma vez obtida, a adequação será considerada verdadeira até que uma nova figuração quebre a correspondência entre formulação e figuração, quando torna-se inaceitável. A busca por adequação, dentro deste trabalho, pode então ser entendida como uma perspectiva genérica para a construção de sistemas através dos quais um especialista no domínio poderia confrontar um modelo formal com resultados experimentais e então fazer uso dos resultados observados para incrementar progressivamente o modelo formal.

A busca por adequação acontece em três níveis: ontológica, epistêmica e heurística. Este processo é inspirado nos três tipos de adequação que foram identificados por McCarthy e Hayes para uma representação de conhecimento [Bittencourt 96]:

- Metafísica: se um mundo construído de acordo com a representação não apresentar contradições com os fatos pertinentes aos aspectos da realidade que desejamos representar.
- Epistemológica: se a representação pode ser utilizada na prática para representar os fatos disponíveis sobre os aspectos de interesse da realidade.
- Heurística: se os processos de raciocínio necessários para a solução dos problemas de interesse podem ser expressos na representação.

Existem diversos exemplos de representações metafisicamente adequadas para expressar conhecimentos científicos ou práticos a respeito de objetos do mundo físico – por exemplo, o modelo atômico de Demócrito, a teoria das *mônadas* de Leibniz, ou o *modelo padrão* da física moderna – que, no entanto, não são epistemologicamente adequadas para representar os conhecimentos necessários para resolver problemas típicos do cotidiano. O contrário, em geral, não ocorre: representações com possibilidades de serem epistemologicamente adequadas para representar o senso comum são, na maioria dos casos, plenamente adequadas metafisicamente. Desta maneira, a adequação metafísica não é um problema central para a IA.

A parte epistemológica da IA estuda que tipos de fatos sobre o mundo estão disponíveis para um observador com determinadas oportunidades para observar, como estes

fatos podem ser representados na memória de um computador e que regras permitem a derivação de conclusões legítimas a partir destes fatos. Em outras palavras, o problema da adequação epistemológica é o problema da correspondência entre mundo externo e a representação propriamente dita.

Uma representação de conhecimento é uma combinação de estruturas de dados e de procedimentos de interpretação que, se usados de maneira correta dentro de um programa, levarão a um comportamento que simule o conhecimento humano. A adequação heurística de uma representação de conhecimento é determinada pela estrutura da representação e eficiência do processo de raciocínio a ela associado.

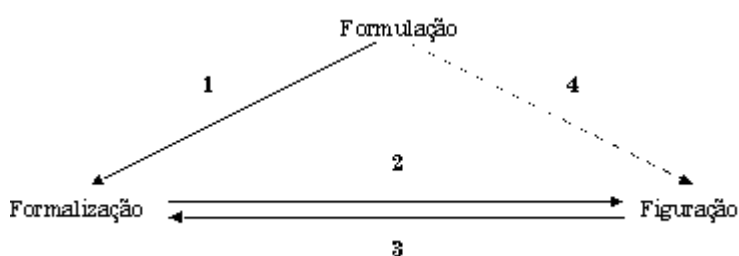


Figura 3.5: O processo de busca por adequação

Adequação ontológica

De acordo com [Nóbrega 00], a adequação ontológica pode ser entendida como a busca, de acordo com os requerimentos do especialista do domínio, de uma evolução satisfatória da ontologia na direção de uma especificação aceitável da conceituação – a busca pelos termos, juntamente com suas relações, necessários para estabelecer uma representação em extensão e uma em intenção para a conceituação corrente. A construção de uma representação em extensão procura determinar quais os termos relevantes, dentro do domínio do instrutor (baseado em sua experiência acumulada). Já a construção de uma representação em intenção tem como objetivo estabelecer as relações entre os termos considerados. A adequação ontológica, abordada neste trabalho, é equivalente a adequação metafísica.

A adequação entre lei e fato é ontológica, pois está-se lidando com o discernimento de mundo do especialista do domínio. A questão a ser respondida neste nível é: “quais os termos a definir?”.

Adequação epistêmica

A adequação ontológica foi concebida como a busca por uma estabilidade em duas

direções opostas: um conceito visto em compreensão (representação em intenção) e em extensão. Uma conceituação deve considerar o significado das relações, que pode não ser apreendido pela representação em extensão – é por isso que a ferramenta teórica aqui considerada incorpora a intervenção do usuário como requerimento para a evolução da ontologia.

O papel da adequação epistêmica é determinar uma coerência entre as representações em extensão e em intenção. A busca por adequação epistêmica pode parecer evidente, mas a adequação ontológica não garante tal restrição. A coerência entre as representações em extensão e em intenção é obtida quando as expressões que diretamente intervêm na construção de uma representação podem ser verificadas quando se constrói a outra.

A adequação entre hipótese e postulado é epistêmica, sendo construída principalmente sobre o conhecimento representado em extensão (os exemplos). A questão a ser respondida neste nível é: “quais as más regras a recusar?”.

Adequação heurística

A adequação epistêmica procura solucionar uma possível incoerência quando se busca por uma adequação ontológica a partir de dois processos independentes. A adequação heurística tem como objetivo resolver, na prática, o problema de coerência da adequação ontológica – a idéia é alcançar a coerência desejada a partir de uma forma mais fácil.

A adequação que se busca entre exemplos e conceito é heurística. Não se sabe a priori quais e quantos exemplos a escolher para se descrever em extensão um conceito de maneira incontestável (tais exemplos variam de acordo com a experiência do especialista do domínio). A questão a ser respondida neste nível é: “quais os bons exemplos a escolher?”.

No capítulo 4 encontra-se uma discussão mais aprofundada acerca do processo de busca por adequação considerada neste trabalho.

3.3.3 Discussão

A abordagem de formação de teorias usando φ -calculus tem sido aplicada ao domínio da Educação Humana, resultando em um ambiente de aprendizagem *web-served* – chamado de PhiInEd – para assistir às fases de *planejamento* e *execução* de um curso [Nó-

brega 02b]. Dentro do servidor PhiInEd, duas fases de um curso são consideradas:

Planejamento. O planejamento de um curso, dentro do PhiInEd, consiste da elaboração de um plano por alguém que o administra, chamado de *professor*. Um *plano* consiste de uma seqüência de *lições*, a serem estudadas por aqueles que seguirão o curso, chamados de *alunos*. O professor pode prover o curso com um número de *recursos* e/ou *exercícios*. Ambos os recursos e exercícios são armazenados dentro do servidor por meio de páginas *web*. Os recursos podem ser vistos como o conteúdo do curso, enquanto que através dos exercícios o professor testa o conhecimento do aluno com relação ao recurso. Uma memória, correspondente ao trabalho do aluno e chamada de *arquitetura de raciocínio* (AR), representa o conhecimento que será usado por um agente artificial ao assistir um agente humano a raciocinar dentro de seu correspondente domínio de conhecimento. Considerando que o servidor PhiInEd provê seus usuários com um agente artificial capaz de (i) desenvolver raciocínio lógico e (ii) aprender a partir de exemplos, mas sem nenhum conhecimento inicial, o trabalho dos alunos, enquanto estudam as lições, consiste em construir o conhecimento do agente (AR) como um resultado da externalização do seu próprio conhecimento. No fim do curso, o sucesso do aluno depende da eficiência do agente assistente que ele construiu.

Execução. Executar um curso dentro do PhiInEd consiste da execução do plano do curso - o estudo pelos alunos das lições, estudo este supervisionado pelo professor. Os alunos elaboram a AR e comunicam-se com o professor, que propõe lições aos alunos, e com os outros colegas, para apresentar sua AR. A execução do curso é organizada como uma seqüência de *diálogos*. Os *assuntos dos diálogos* correspondem aos títulos das lições do plano - se um curso tem três lições, então serão executados três diálogos, um para cada lição. Um diálogo é composto por uma seqüência de *mensagens*, e o *assunto da mensagem* é, tal como para o diálogo, o título da lição. O professor pode declarar uma lição como aprendida por um aluno quando o diálogo sobre aquela lição tem no mínimo as seguintes duas mensagens: (i) do professor para o aluno, transportando as lições a serem estudadas, e (ii) do aluno para o professor, transportando a AR do aluno com respeito à ao assunto da lição do diálogo. Ao lado destas duas mensagens, duas outras podem ocorrer em um diálogo, no caso em que o aluno solicita explicações enquanto estuda a lição.

O objeto de estudo para o curso foi Condições Gerais de Venda (CGV). A definição de CGV requer um entendimento do processo de formação de um contrato. É estabelecido que sempre que uma oferta tem uma aceitação resulta um acordo que restringe o com-

portamento que os dois lados devem adotar mutuamente. Neste contexto, CGV são definidas como uma oferta de contrato feita por um vendedor a um comprador interessado em adquirir o produto. Esta oferta constitui então a norma individual do comportamento que o vendedor impõe a seus compradores. A simples adesão incondicional das condições de um vendedor por um comprador deveria ser suficiente para formar o contrato, e a norma individual composta pelas condições do vendedor torna-se então a norma comum para os dois lados.

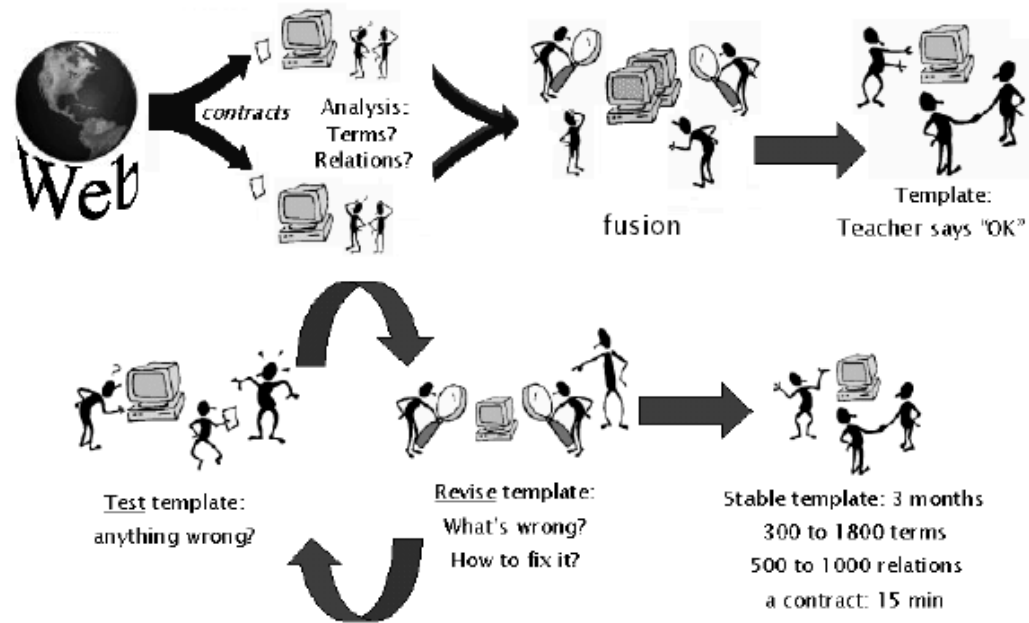


Figura 3.6: O cenário de experimentação.

O cenário de experimentação é ilustrado na figura 3.6. O professor tem inicialmente estabelecido as CGV como o assunto de estudo. Uma AR deveria então ser construída para esta classe de contrato. A meta era obter uma simples AR como resultado do trabalho do grupo inteiro. Primeiramente, os alunos eram distribuídos em sete grupos de trabalho, cada grupo trabalho em um única máquina. Cada grupo deveria então preparar uma AR para analisar um documento CGV da *web*. Logo após, uma simples AR deveria ser gerada como um fusão de ARs dos grupos. Finalmente, este AR resultante deveria ser revisada sob a supervisão do professor, para ser considerado pelo grupo como estável.

3.4 Conclusão

ϕ -calculus é uma ferramenta teórica para a construção interativa e racional de teorias [Nóbrega 00, 01a, 01b]. As atitudes mentais matemáticas são usadas para realizar a categorização natural. Elas, por sua vez, pertencem a um dos processos computacionais da

ferramenta: a *expressão, formalização e computação*. A criação de representação mental, a formalização da construção de conhecimento e a interação entre instrutor humano e sistema (artificial) são mecanismos contidos em φ -calculus.

No contexto do raciocínio assíncrono interativo, o agente humano (usuário e especialista do domínio) dispõe de um agente artificial aprendiz como um colaborador – dada sua habilidade para desenvolver computações – para ajudá-lo a raciocinar dentro de um domínio em estudo, refinando sua representação mental. As atividades de aprendizagem e supervisão, do usuário para o agente aprendiz, incluem *ensino, avaliação* da qualidade do conhecimento e *correção* do mesmo. A ferramenta permite ao usuário formar um conceito, dentro de um domínio de seu interesse, pela supervisão da evolução de uma ontologia. O programa WebContract + Praal (WCP) é usado como demonstrador da metodologia (ver anexo 1).

Fazendo um paralelo entre as abordagens aqui consideradas, algumas limitações do projeto SAID, frente ao projeto φ -calculus, puderam ser detectadas:

- A interação entre o agente SAID e o especialista do domínio é feita apenas a partir das argumentações do agente – o instrutor não tem acesso direto ao conhecimento armazenado pelo agente.
- A idéia de *adequação* usada no SAID não é bem definida: "critério a ser satisfeito, um critério de aprendizagem, que estabelece o que significa uma melhor hipótese em uma dada amostra".
- No SAID, o especialista do domínio envia pares <problema, solução> para seu agente aprendiz. Em φ -calculus, o especialista não precisa conhecer à priori a solução para o problema, pois este também está numa situação de descoberta - tanto é que o agente racional além de aprendiz é colaborador.
- A interação entre o especialista do domínio e o agente é mais direta e simples no φ -calculus.

Apesar de SAID constituir-se numa evolução para φ -calculus, infelizmente o problema do *silêncio* ainda não foi resolvido (como será visto adiante). No próximo capítulo é apresentada a concepção de um agente racional baseado em φ -calculus (raciocínio assíncrono interativo).

Capítulo 4

Um sistema de construção interativa de teorias

4.1 Introdução

Como abordado no capítulo 2, a intervenção do usuário tem um papel importante no processo de descoberta científica feita por entidades artificiais. Neste trabalho em especial, a idéia geral é discutir a importância da interação instrutor-aprendiz na concepção de teorias feitas por um agente racional cuja arquitetura é baseada em raciocínio assíncrono interativo. No capítulo 3 foi visto que a ferramenta teórica ϕ -calculus, cuja base é o raciocínio assíncrono interativo, possui mecanismos para a criação de representação mental, para a formalização da construção de conhecimento e para a interação entre instrutor humano e sistema artificial.

O modelo genérico de agente aprendiz proposto aprende tanto por intervenção direta do instrutor, como a partir de exemplos. A *aprendizagem a partir de exemplos* é um dos principais tipos de sistemas indutivos. O sistema induz uma hipótese a partir de um conjunto de objetos pré-classificados por um instrutor. Formalmente, um exemplo é um par $(x, f(x))$, onde x é a entrada e $f(x)$ é saída da função aplicada a x . Dada uma coleção de exemplos de f , a tarefa da indução é: retornar uma função h que aproxima f . A função h é chamada de *hipótese* [Russell 95]. Esta hipótese será testada e julgada segundo um critério de adequação, feita em três níveis: ontológico, epistêmico e heurístico.

Este capítulo está estruturado em três seções. Na seção 4.2, é discutida a fundamentação teórica do raciocínio assíncrono interativo. Na seção 4.3, são apresentadas as três fases do ciclo/processo de modelagem: ensino, avaliação e correção. Por fim, na seção 4.4, é feito um paralelo entre este trabalho e outros trabalhos em agente aprendizes.

4.2 O raciocínio assíncrono interativo

Como abordado no capítulo anterior, o raciocínio assíncrono interativo, que envolve uma relação de ensino/aprendizagem entre um instrutor humano e um agente artificial, pode ser definido/resumido da seguinte maneira: um instrutor humano, com conhecimento sobre um dado assunto/domínio, deseja modelar este conhecimento com a ajuda de um agente artificial no papel de aprendiz e colaborador. A intenção do instrutor (ou especialista do domínio) não é apenas modelar este conhecimento, mas também verificar sua veracidade/amplitude através da observação do comportamento do agente frente a situações de teste do conhecimento. Caso seja necessário, reorganiza-se o conhecimento e repete-se o processo enquanto o instrutor julgar oportuno. Este processo de colaboração/aprendizagem entre o especialista do domínio e o agente artificial é assíncrono e interativo.

Uma proposta de modelo genérico de agente aprendiz, baseado em raciocínio assíncrono interativo, é ilustrada na figura 4.1. Este agente é composto de três componentes conceituais:

- **Componente de aprendizagem.** Responsável pela extração de regras/regularidades dos objetos que compõem a amostra de exemplos. O instrutor usa estas regras para criar uma hipótese a ser testada.

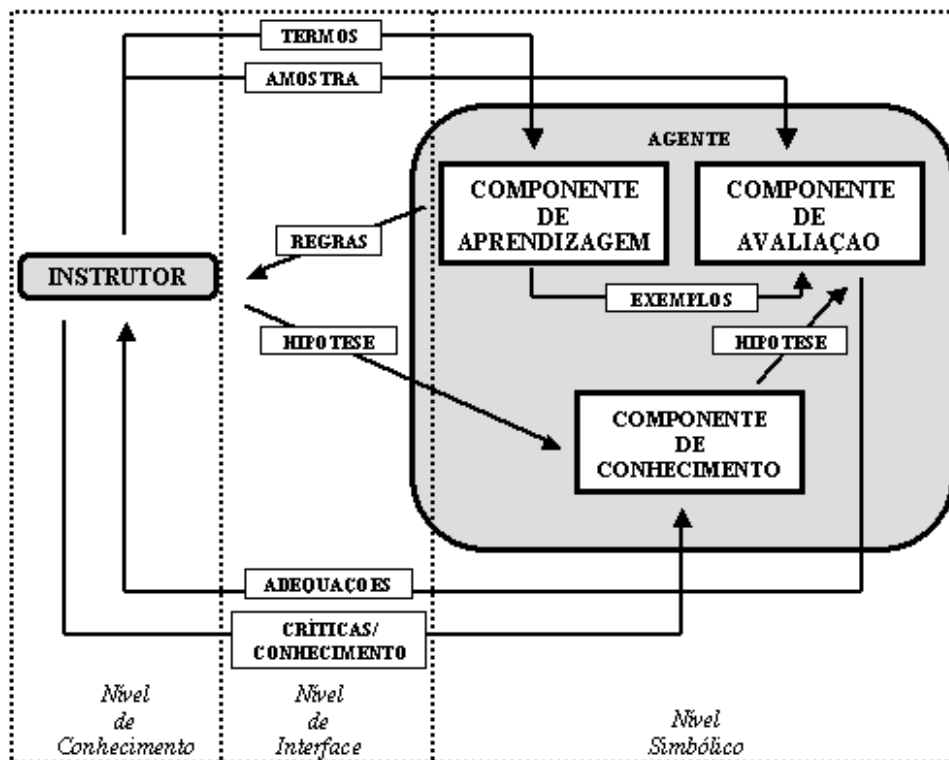


Figura 4.1: Agente aprendiz baseado em raciocínio assíncrono interativo.

- **Componente de avaliação.** Ao receber do instrutor uma nova amostra de exemplos, este componente avalia se a hipótese, ao recebe-la do componente de conhecimento, é considerada adequada para representar o conceito que o instrutor deseja obter. O componente de avaliação também recebe do componente de aprendizagem os exemplos usados como a amostra da qual se extraiu as regras que compõem a hipótese, sendo testada, para avaliar se esta é adequada como amostra de aprendizagem. Este componente envia para o instrutor uma resposta da avaliação na forma de adequações (melhor abordadas posteriormente).
- **Componente de conhecimento.** Responsável por armazenar a hipótese que está sendo testada.

Neste modelo, o instrutor (ou especialista do domínio) apresenta ao agente aprendiz um conjunto de *termos*, para ser usado como restrição/critério para a apresentação de uma *amostra de exemplos* (positivos e negativos¹), a partir da qual o agente aprenderá, e de uma amostra de exemplos para a avaliação da hipótese. O *componente de aprendizagem* é responsável pela extração de *regras*, que representam em intenção os exemplos, apresentando-as ao instrutor, para que ele as organize em conjecturas. Em seguida, o instrutor escolhe uma das conjecturas como a *hipótese* a ser testada e a armazena no *componente de conhecimento*. Para verificar se a hipótese escolhida é adequada ou não, o instrutor apresenta uma nova amostra, chamada aqui de *amostra de avaliação*, ao seu agente aprendiz e espera pelo resultado. Este resultado, aqui chamado de *adequações*, será julgado pelo instrutor. Após este julgamento, o instrutor emite *críticas* ao comportamento de seu agente, sugerindo modificações na hipótese, ou a escolha de uma nova hipótese a partir das conjecturas anteriormente organizadas por ele. O agente pode então ser avaliado novamente ou o especialista do domínio pode recomeçar todo o processo de aprendizagem.

Uma questão a ser observada é a possibilidade do especialista do domínio apresentar diretamente o *conhecimento* que ele deseja modelar ao seu agente, abrindo mão, assim, da apresentação de uma amostra de exemplos e da aprendizagem automática. Logo após, a avaliação e a correção podem ser feitas. No componente de avaliação, quando da avaliação do comportamento do agente frente aos exemplos da amostra, a hipótese corrente é consultada.

Ainda de acordo com o modelo, a interação instrutor-agente pode ser representada

¹ Um exemplo negativo é o mesmo que um contra-exemplo.

em três níveis (ainda na figura 4.1):

- **Nível de conhecimento.** Refere-se ao conhecimento do instrutor acerca do domínio a ser tratado/modelado, bem como a seu discernimento do mundo. É o que o especialista do domínio entende/rejeita/aceita a partir da interação com seu agente aprendiz.
- **Nível de interface.** É o veículo que possibilita um formalismo/linguagem comum ao instrutor e seu agente aprendiz para a comunicação/interação/colaboração entre eles.
- **Nível simbólico.** Diz respeito à representação interna do conhecimento no agente aprendiz. Esta representação interna não é de interesse do especialista do domínio.

A interação instrutor-agente consiste de um processo de modelagem do conhecimento detido pelo especialista do domínio, e que ele procura transmitir a seu agente aprendiz/colaborador.

4.3 O processo de modelagem

Como citado no capítulo 1, modelo neste trabalho é usado como sinônimo de imagem reproduzida pelo especialista do domínio acerca de seu próprio conhecimento, mantendo uma correspondência com os aspectos da realidade que fazem sentido para o especialista. O ciclo/processo de modelagem compreende três fases: *ensino*, *avaliação* e *correção*. Na primeira, o instrutor organiza os termos relevantes do domínio e insere uma amostra de exemplos e contra-exemplos para que o agente aprenda a partir dela. O instrutor pode ensinar seu agente através da inserção direta o conhecimento, sem passar pela aprendizagem de máquina. Após a fase de *ensino*, começa a fase de *avaliação do conhecimento*, quando o especialista do domínio apresenta uma nova amostra ao agente, para verificar se o que ele aprendeu é satisfatório. Depois, na fase de *correção*, há o refinamento do conhecimento aprendido, quando o especialista do domínio critica o conhecimento do agente. A pergunta-chave de todo processo de modelagem é: qual a utilidade de se dispor de um agente artificial, ensiná-lo algo e tê-lo como colaborador? O instrutor responderá a esta questão de acordo com as suas perspectivas.

O especialista do domínio apresenta os termos relevantes ao agente – estes termos servem como critério segundo o qual a aprendizagem se dará. Em seguida, uma amostra composta de exemplos e contra-exemplos é também apresentada. No *componente de aprendizagem*, um conjunto de regras (que caracterizam a amostra) é extraído a partir dos exemplos positivos e negativos. O instrutor, de posse destas regras, as organiza em uma

hipótese (não é necessário considerar todas as regras obtidas) que representará o conceito a ser avaliado. É importante notar que este conceito ainda não é considerado adequado (pelo instrutor) para representar a amostra em intenção. Por isso, nesse momento, ainda é chamado de hipótese. A hipótese será armazenada pelo *componente de conhecimento*, representando a informação detida no momento pelo agente. Assim termina a fase de *ensino* e começa a de *avaliação*.

Na fase de *avaliação*, após a apresentação dos termos, uma nova amostra é apresentada ao agente com a intenção de verificar se o comportamento deste é satisfatório; em outras palavras, verificar se a hipótese detida pelo agente é adequada. Esta verificação é feita no *componente de avaliação*, em colaboração com os *componentes de conhecimento e aprendizagem*. Por fim, o agente mostra a seu instrutor o resultado, na forma de um processo de busca por adequação (ontológica, epistêmica e heurística). De posse do resultado, o instrutor o avalia como:

- **Satisfatório.** O comportamento do agente é o esperado. O ciclo/processo de modelagem é concluído.
- **A hipótese não é adequada.** É necessário escolher uma nova hipótese a ser testada, obtida a partir das regras geradas na fase de *ensino*. Quando a hipótese foi passada diretamente pelo instrutor ao agente, em que não foi usada nenhuma amostra de aprendizagem, o ciclo de modelagem é refeito.
- **A hipótese testada precisa de algumas modificações.** O instrutor exclui regras da hipótese, que é armazenada no *componente de conhecimento*. Depois de feitas as modificações, o instrutor pode, opcionalmente, voltar para a fase de *avaliação*.
- **A amostra de ensino e/ou os termos foram mal escolhidos.** O processo de aprendizagem tem que ser refeito.

Assim que recebe o resultado, termina a fase de *avaliação* e começa a fase de *correção*, quando o instrutor criticará e colocará em prática a decisão que tomou com relação à avaliação da hipótese. Assim, o ciclo/processo de modelagem é concluído.

Como abordado no capítulo 3, os trabalhos sobre construção interativa de teorias apóiam-se em três vertentes complementares – conceitual, formal e experimental. O ciclo de modelagem é abordado segundo estas vertentes.

4.3.1 Vertente conceitual

A vertente conceitual tem como meta principal investigar a exploração da racionalidade para possibilitar coerência e ausência de ambigüidade durante o processo de explicitação do conhecimento. Nesta vertente é mostrada as bases conceituais que sustentam o ciclo de modelagem, ou a interação instrutor-agente, analisando segundo as fases de ensino, avaliação e correção. Também é explicitado o processo de busca por adequação (ontológica, epistêmica e heurística), definido no capítulo 3, aplicado ao modelo genérico de agente aprendiz proposto neste trabalho.

4.3.1.1 Fase de ensino

Como discutido no capítulo 3, ϕ -calculus é uma ferramenta teórica para a categorização de *atitudes mentais matemáticas*, que ajudam um agente a racionalizar a construção de conhecimento. As atitudes mentais matemáticas (AMMs) são em número de doze: *lei, fato, hipótese, postulado, exemplo, conceito, lema, conjecturas, princípio, opinião, objeção e prova*. Na fase de ensino do ciclo de modelagem, as AMMs consideradas são as dez primeiras.

A fase de ensino do ciclo pode ser feita de duas formas: *indireta*, quando o especialista do domínio deseja que seu agente aprenda a partir de uma amostra (composta de exemplos positivos e negativos, como dito anteriormente) e extraia a partir daí um conceito que a represente em intenção, e *direta*, quando o instrutor decide apresentar diretamente o conhecimento que deseja modelar sem passar pela aprendizagem automática. Na figura 4.2 é ilustrada a fase de *ensino*².

O ensino indireto acontece da seguinte forma (figura 4.2a): o instrutor, a partir de seu conhecimento e discernimento do domínio de aplicação, decide quais são os termos relevantes como critério/restrrição para a inserção da amostra (seta 1). Feito isto, apresenta-os ao agente. Em seguida, apresenta ao aprendiz³ uma amostra, composta de exemplos e contra-exemplos do conceito a ser aprendido (seta 2). No *componente de aprendizagem* são extraídas regras, aqui chamadas de lemas (seta 3), que são organizadas (pelo especia-

² A hipótese (vinda do ensino direto ou indireto) é compartilhada entre os componentes de avaliação e conhecimento (onde é armazenada).

³ lembrando que o agente na interação é considerado como um aprendiz e um colaborador pelo especialista do domínio, já que este também está em uma situação de descoberta – o instrutor pode não ter ainda uma visão clara acerca de seu próprio conhecimento, desejando experimentá-lo com a ajuda de uma entidade artificial (um agente).

lista do domínio) em conjuntos, aqui chamados de conjecturas (seta 4). Uma dessas conjecturas é escolhida como a hipótese que representará o conceito, a ser avaliada (seta 5). Quando a hipótese é armazenada no *componente de conhecimento*, começa a fase de *avaliação* (seta 6). Em síntese, *exemplos* → *lemas* → *conjecturas* → *hipótese* (que é formada por um conjunto de *postulados*).

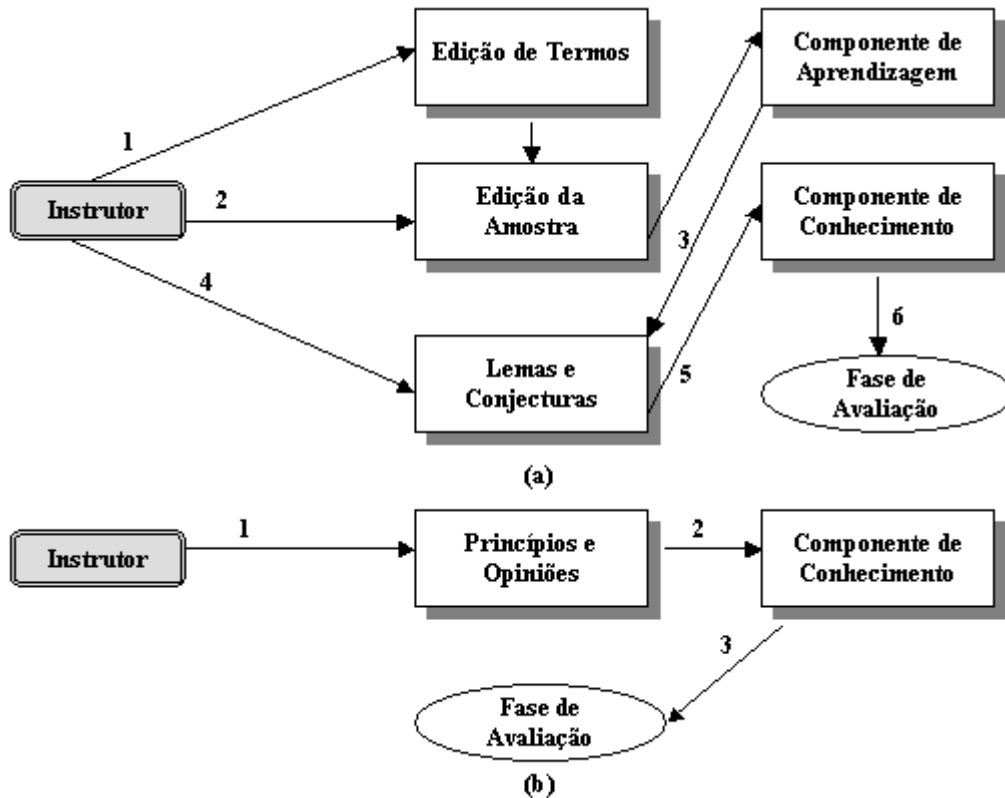


Figura 4.2: A fase de ensino: (a) indireto e (b) direto.

No ensino direto (figura 4.2b), o especialista do domínio apresentará o conhecimento diretamente ao seu agente aprendiz, observando um conjunto de fatos, que respeitam leis que obedecem ao discernimento do instrutor com relação ao domínio. Estes fatos para serem apresentados ao agente, serão decompostos em princípios⁴ que, posteriormente, irão compor opiniões⁵ (seta 1). Tal como os lemas e conjecturas, o instrutor pode estruturar os princípios em opiniões e escolher uma dela para representar a hipótese a ser testada (seta 2). Terminado isto, dá-se início a fase de *avaliação* (seta 3). Em síntese: *fato* → *lei* → *princípio* → *opinião* → *hipótese* (que é formada por um conjunto de *postulados*).

⁴ Princípios são regras obtidas diretamente do conhecimento vindo do instrutor – no caso do ensino direto.

⁵ Opinião é um conjunto de princípios.

4.3.1.2 Fase de avaliação

Na figura 4.3 é ilustrada a fase de *avaliação*. Terminada a fase de *ensino*, o instrutor apresenta os termos (seta 1), que também foram apresentados na fase de ensino, e uma nova amostra⁶ (seta 2) para testar a corretude/amplitude do conhecimento armazenado pelo seu agente – que neste momento é compartilhado entre *componente de avaliação* e *componente de conhecimento* (seta 3). O *componente de aprendizagem* compartilha a amostra de aprendizagem com o componente de avaliação, para que esta seja avaliada como adequada ou não para a extração de regras que, posteriormente, irão compor a hipótese. O resultado é enviado ao especialista do domínio (seta 5). Em seguida, começa a fase de *correção* (seta 6). O objetivo da fase de avaliação para o especialista do domínio é decidir se o conhecimento apreendido pelo agente artificial corresponde ao conhecimento que ele pretendia passar na fase de ensino.

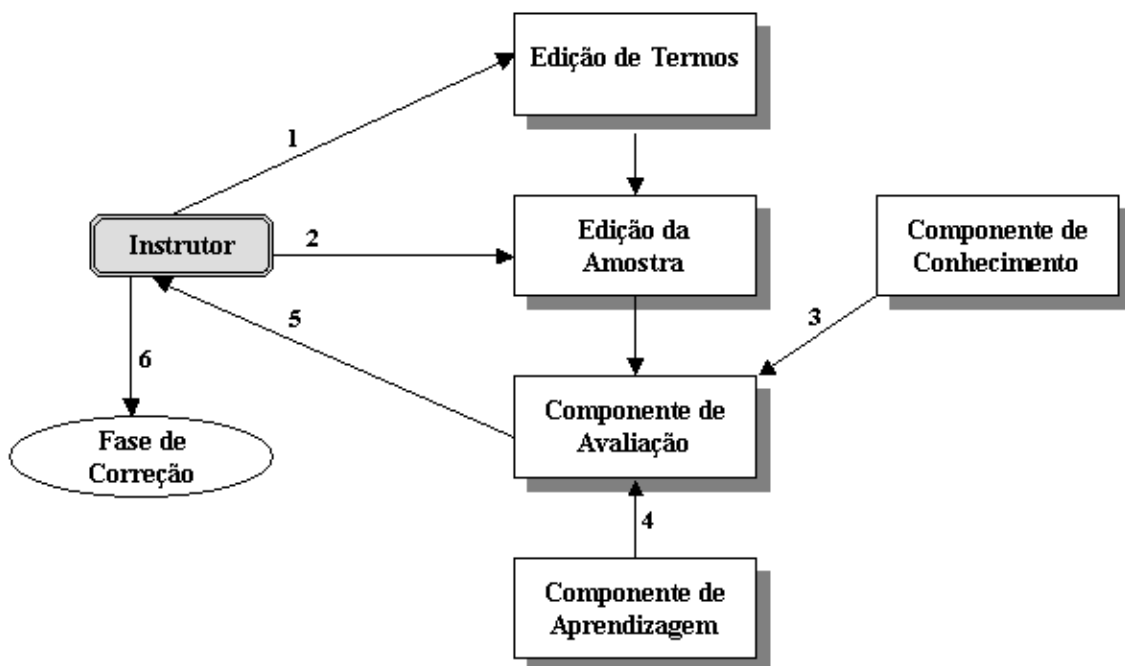


Figura 4.3: A fase de avaliação.

Como abordado no capítulo 3, a avaliação de um agente aprendiz por um especialista do domínio é feita através de uma busca por adequação em três níveis: ontológico, epistêmico e heurístico. A adequação ontológica alerta quando a indefinição de alguns termos usados em objetos da amostra de avaliação, que não aparecem na descrição intencional do conhecimento – quando o agente diz ao instrutor se todos os termos são definidos. A pergunta neste nível de adequação é: quais os termos a definir/escolher? A adequa-

⁶ Os objetos desta amostra podem ser considerados como *objetos experimentais*.

ção epistêmica alerta quanto ao uso incorreto de termos, em objetos da amostra de avaliação, que não correspondem aos postulados da hipótese corrente – quando o agente diz ao instrutor se os exemplos da amostra de avaliação estão de acordo com a descrição em compreensão que ele mantém. A pergunta é: quais as más regras a recusar? Lembrando que as regras podem ser lemas (extraídos no ensino indireto) ou princípios (extraídos no ensino direto). Por fim, a adequação heurística verifica se a amostra de exemplos, positivos ou negativos, usada como amostra de aprendizagem é adequada para servir para a geração dos postulados que constituem a hipótese sendo testada. No capítulo 3, foi visto que a adequação heurística avalia a eficiência do processo de raciocínio associado a representação do conhecimento, o que inclui o raciocínio por *generalização* para a geração de conceitos a partir de exemplos e contra-exemplos (é apresentado a um programa uma série de exemplos e contra-exemplos, e o programa deve gerar uma representação, em uma linguagem formal dada, do conceito mais simples capaz de explicar as informações disponíveis [Winston 75]). A escolha de exemplos adequados para compor a amostra de aprendizagem se torna crucial para o sucesso do processo de modelagem, no caso do ensino indireto. Quando o ensino é direto, não existe uma amostra de aprendizagem, a resposta para o nível de adequação heurística é *não verificado*. A pergunta-chave neste nível é: quais os bons exemplos a escolher? Na fase de avaliação, o processo de busca por adequação deve ser feito objeto a objeto – a resposta nos três níveis (ontológico, epistêmico e heurístico) é dada para cada objeto experimental separadamente.

É importante lembrar que o resultado enviado ao instrutor pelo agente, após a avaliação da amostra, apenas indica uma inadequação/adequação nos três níveis, servindo simplesmente como uma orientação. A palavra final, acerca de possíveis mudanças na hipótese corrente armazenada pelo agente ou nos termos, cabe apenas ao instrutor. A intuição do especialista do domínio é usada como medida de desempenho⁷.

4.3.1.3 Fase de correção

Na figura 4.4 é ilustrada a fase de *correção*. Após avaliar o resultado, o instrutor decide modificar (ou criticar) a hipótese sendo testada (seta 1), armazenada no componente de conhecimento, escolher uma nova hipótese para avaliar (setas 2 e 3), a partir das conjecturas organizadas anteriormente pelo instrutor na fase de ensino, voltando para a fase de

⁷ Medida de desempenho (MD) é o critério (medida que deve ser imposta pelo exterior) que determina o grau de sucesso de um agente [Russell 95]. É importante ressaltar que a má escolha da MD pode acarretar comportamento indesejado. Resta o problema de saber quando avaliar o desempenho.

avaliação (seta 4), ou recomeçar todo o ciclo/processo de modelagem. Quando o instrutor decide modificar a hipótese que está sendo testada, ou escolher uma nova hipótese, a hipótese modificada/escolhida constitui-se numa objeção⁸ e seus postulados numa prova⁹. Logo em seguida, a objeção se tornará a hipótese corrente do agente, a ser testada. Em síntese: *hipótese* → *objeção* (composta por um conjunto de *provas*) → *hipótese* (composta por um conjunto de postulados).

Algumas considerações acerca do ciclo de modelagem se faz necessário: na fase de ensino é importante verificar que na amostra de aprendizagem não podem existir dois objetos com características iguais, mas com classificação diferente (um exemplo é positivo e o outro é negativo). Isto pode gerar regras contraditórias (dependendo do algoritmo de aprendizagem), gerando um conhecimento ambíguo. Uma outra questão é sobre a *legibilidade* das regras (lemas) que são geradas por aprendizagem automática. Estas regras são manipuladas pelo instrutor para formar a hipótese a ser testada, por isso devem ter um formato que possa ser compreendido sem problemas pelo especialista do domínio.

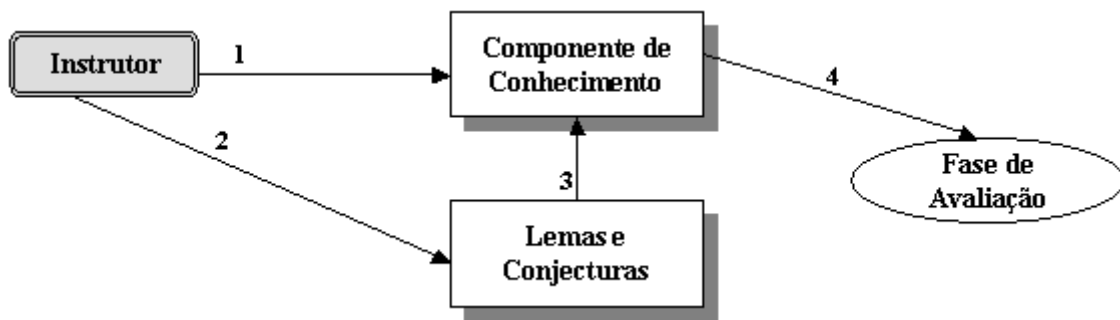


Figura 4.4: A fase de correção.

Todas as fases do ciclo de modelagem (ensino, avaliação e correção) são obrigatórias, sendo que o instrutor pode, opcionalmente, retornar a fase de avaliação após a fase de correção. Apesar de que um ciclo de modelagem pode ser seguido por outros ciclos, os ciclos seguintes devem ter um ensino direto, ou seja, o primeiro ciclo pode ter ensino indireto, mas a partir do segundo, o ensino deve ser direto. Isto se deve ao fato de que os exemplos (positivos ou negativos) da amostra de aprendizagem de um ciclo podem "entrar em conflito" com os exemplos de um outro ciclo, pois, como os exemplos de uma amostra de aprendizagem são irrefutáveis, objetos com características iguais e classificação diferente podem vir a ser considerados pelo instrutor, mesmo que pertençam às amostras de aprendizagem distintas. Como as amostras podem ser grandes (podem conter muitos exem-

⁸ Objeção diz respeito à discordância do especialista do domínio com relação à hipótese corrente, apreendida pelo agente aprendiz, que provoca inadequações em algum dos níveis.

plos), a tarefa de escolher exemplos adequados pode acabar fugindo ao controle e discernimento do instrutor. Quando o ciclo de modelagem em que o ensino foi indireto é seguido por outro ciclo, em que o ensino é direto, a resposta da adequação heurística é diferente de *não verificado*, pois existe à priori uma amostra de aprendizagem.

Com relação à associação do conhecimento às atitudes mentais matemáticas, o processo/ciclo de modelagem pode ser visto segundo um ciclo que envolve os três processos da metodologia de φ -calculus: expressão, formalização e computação. A expressão equivale a fase de ensino, a formalização a fase de correção e a computação a fase de avaliação. Na figura 4.5 está ilustrada esta idéia.

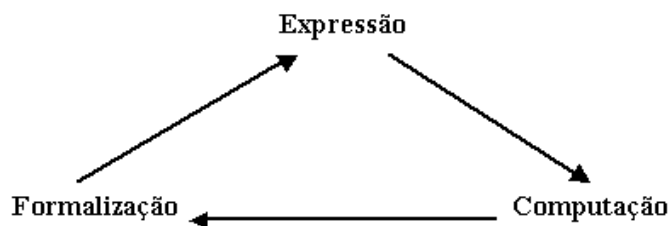


Figura 4.5: Ciclo de modelagem segundo os processos da metodologia do φ -calculus.

4.3.2 Vertente formal

A interação instrutor-agente agora é analisada segundo a vertente formal. Uma definição formal é necessária por duas razões específicas: (i) é precisa e (ii) provê notação, e boas notações ajudam-nos a pensar e expressar nossas idéias de maneira mais clara [Sipser 97]. Para obter uma definição formal do ciclo de modelagem, foi usado um tipo de modelo computacional chamado autômato pushdown, ou autômato de pilha (definição, anexo B).

Embora a representação por autômatos não seja epistemologicamente adequada para a representação de comportamentos inteligentes complexos (mesmo um processo simples necessitaria de um número astronômico de estados, e além disto estes estados seriam em geral desconhecidos), sua simplicidade, porém, permite definir de maneira clara o que poderia significar "poder" e "acreditar" em relação a um programa, além de ressaltar que tal significado depende tão só do mundo externo (o sistema global de autômatos) e das possíveis interações entre seus componentes, mas principalmente da delimitação do subsistema a ser considerado [Bittencourt 96].

O autômato de pilha correspondente ao modelo genérico de agente aprendiz base-

⁹ Provas são as regras que compõem e sustentam a objeção.

ado em raciocínio assíncrono interativo, apresentado na figura 4.1, é mostrado na figura 4.6.

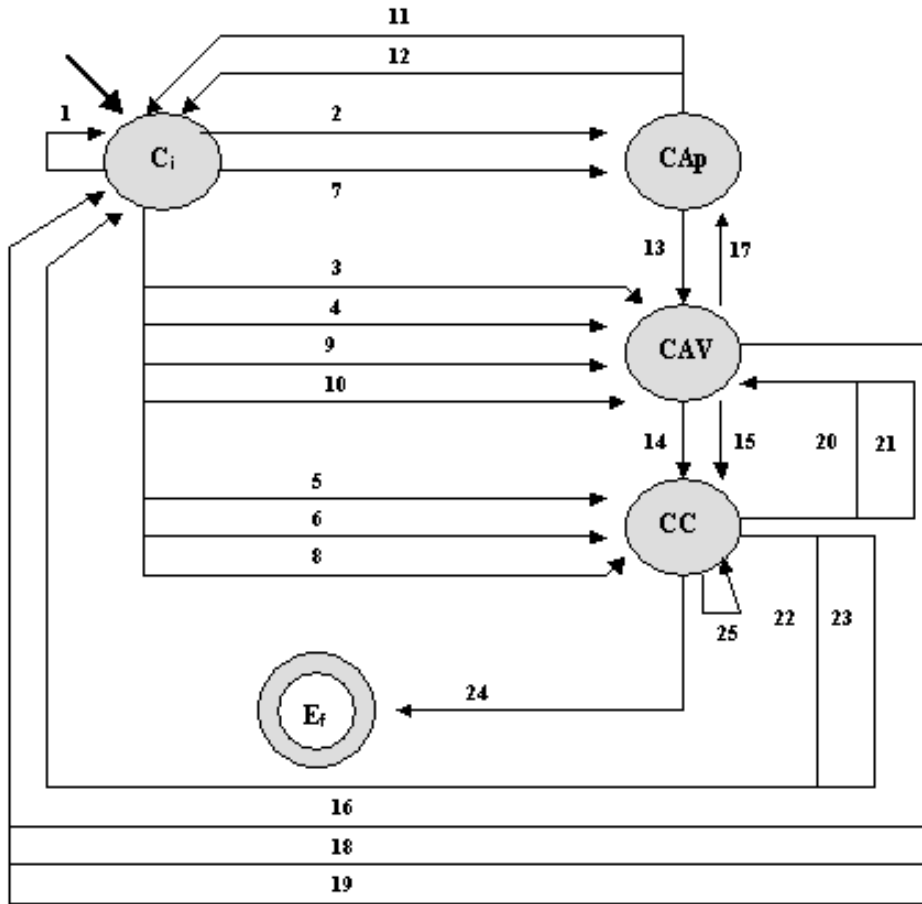


Figura 4.6: Autômato representando o modelo genérico de agente aprendiz.

O autômato de pilha é uma 6-tupla: o conjunto de estados é: $Q = \{C_i, CAp, CAv, CC, E_f\}$. O alfabeto de entrada é: $\Sigma = \{(\text{termos}), (\text{amostra}_{ap}), (\text{regras}), (\text{hipótese}), (\text{amostra}_{av}), (\text{conhecimento}), (\text{adequações}) \text{ e } (\text{críticas})\}$. O alfabeto da pilha é: $\Gamma = \{\&, \$, (\text{termos}), (\text{amostra}_{ap}), (\text{regras}), (\text{hipótese}), (\text{hipótese}_i), (\text{hipótese}_i), (\text{hipótese}_{i1}), (\text{amostra}_{av}), (\text{amostra}_{avi}), (\text{adequações}) \text{ e } (\text{críticas})\}$. O estado inicial é: $q_0 = \{C_i\}$. O conjunto de estados de aceitação: $F = \{E_f\}$. Por fim, a função de transição é:

- $\delta_1: (C_i, \epsilon, \epsilon) \rightarrow (C_i, \$)$
- $\delta_2: (C_i, (\text{termos}), \$) \rightarrow (CAp, (\text{termos}))$
- $\delta_3: (C_i, (\text{amostra}_{av}), (\text{hipótese}_i)) \rightarrow (CAv, (\text{amostra}_{av}))$
- $\delta_4: (C_i, (\text{amostra}_{av}), (\text{hipótese})) \rightarrow (CAv, \&)$
- $\delta_5: (C_i, (\text{conhecimento}), \$) \rightarrow (CC, (\text{hipótese}))$
- $\delta_6: (C_i, (\text{críticas}), (\text{adequações})) \rightarrow (CC, (\text{críticas}))$
- $\delta_7: (C_i, (\text{amostra}_{ap}), (\text{termos})) \rightarrow (CAp, (\text{amostra}_{ap}))$
- $\delta_8: (C_i, (\text{hipótese}), (\text{regras})) \rightarrow (CC, (\text{hipótese}_i))$
- $\delta_9: (C_i, (\text{termos}), (\text{hipótese})) \rightarrow (CAv, (\text{hipótese}))$
- $\delta_{10}: (C_i, (\text{termos}), (\text{hipótese}_i)) \rightarrow (CAv, (\text{hipótese}))$

- $\delta_{11}: (CAp, \varepsilon, (\text{termos})) \rightarrow (Ci, (\text{termos}))$
- $\delta_{12}: (CAp, (\text{regras}), (\text{amostra}_{ap})) \rightarrow (Ci, (\text{regras}))$
- $\delta_{13}: (CAp, (\text{amostra}_{ap}), (\text{amostra}_{avi})) \rightarrow (CAv, (\text{amostra}_{avi}))$
- $\delta_{14}: (CAv, \varepsilon, (\text{amostra}_{avi})) \rightarrow (CC, (\text{amostra}_{avi}))$
- $\delta_{15}: (CAv, \varepsilon, \&) \rightarrow (CC, \&)$
- $\delta_{16}: (CAv, (\text{adequações}), (\text{hipótese})) \rightarrow (Ci, (\text{adequações}))$
- $\delta_{17}: (CAv, \varepsilon, (\text{amostra}_{av})) \rightarrow (CAp, (\text{amostra}_{avi}))$
- $\delta_{18}: (CAv, \varepsilon, (\text{hipótese}_i)) \rightarrow (Ci, (\text{hipótese}_{i1}))$
- $\delta_{19}: (CAv, \varepsilon, (\text{hipótese})) \rightarrow (Ci, (\text{hipótese}_1))$
- $\delta_{20}: (CC, (\text{hipótese}), (\text{amostra}_{avi})) \rightarrow (CAv, (\text{hipótese}))$
- $\delta_{21}: (CC, (\text{hipótese}), \&) \rightarrow (CAv, (\text{hipótese}))$
- $\delta_{22}: (CC, \varepsilon, (\text{hipótese})) \rightarrow (Ci, (\text{hipótese}))$
- $\delta_{23}: (CC, \varepsilon, (\text{hipótese}_i)) \rightarrow (Ci, (\text{hipótese}_i))$
- $\delta_{24}: (CC, \varepsilon, (\text{críticas})) \rightarrow (Ef, \varepsilon)$
- $\delta_{25}: (CC, \varepsilon, (\text{críticas})) \rightarrow (CC, (\text{hipótese}))$

A linguagem reconhecida pelo autômato da figura 4.6 é: $x = (\text{termos}) (\text{amostra}_{ap}) (\text{regras}) (\text{hipótese}) (\text{termos}) (\text{amostra}_{av}) (\text{amostra}_{ap}) (\text{hipótese}) (\text{adequações}) (\text{críticas})$, $y = (\text{conhecimento})$ e $z = (\text{termos}) (\text{amostra}_{av}) (\text{hipótese}) (\text{adequações}) (\text{críticas})$, então, $L_{MG} = \{xz^m \text{ ou } yz^n, \text{ tal que } m \geq 0 \text{ e } n > 0\}$. Significa que o instrutor (estado Ci) envia os *termos* para o componente de aprendizagem (estado CAp), logo em seguida, envia a amostra de aprendizagem (*amostra_{ap}*) também para o mesmo componente. A partir desta amostra, é extraído um conjunto de *regras*, que é apresentado ao instrutor, sendo por este organizado em uma *hipótese*, que é armazenada no componente de conhecimento (estado CC). Antes de testar a hipótese, o instrutor envia os *termos* para o componente de avaliação (estado CAv), depois a amostra de avaliação (*amostra_{av}*), recebe do componente de aprendizagem a amostra que foi usada para a extração de regras (*amostra_{ap}*) e recebe do componente de conhecimento a *hipótese* corrente. Após a avaliação, é apresentado o resultado (as *adequações*) ao instrutor, que por sua vez envia *críticas*. O instrutor pode, opcionalmente, enviar seu *conhecimento* diretamente, sem usar nenhuma amostra de aprendizagem. Nas duas situações, a avaliação da hipótese corrente pode ser refeita após a correção da mesma, ou o ciclo de modelagem é finalizado (estado Ef).

4.3.2.1 Fase de ensino

O autômato de pilha referente a fase de ensino do ciclo de modelagem é mostrado na figura 4.7.

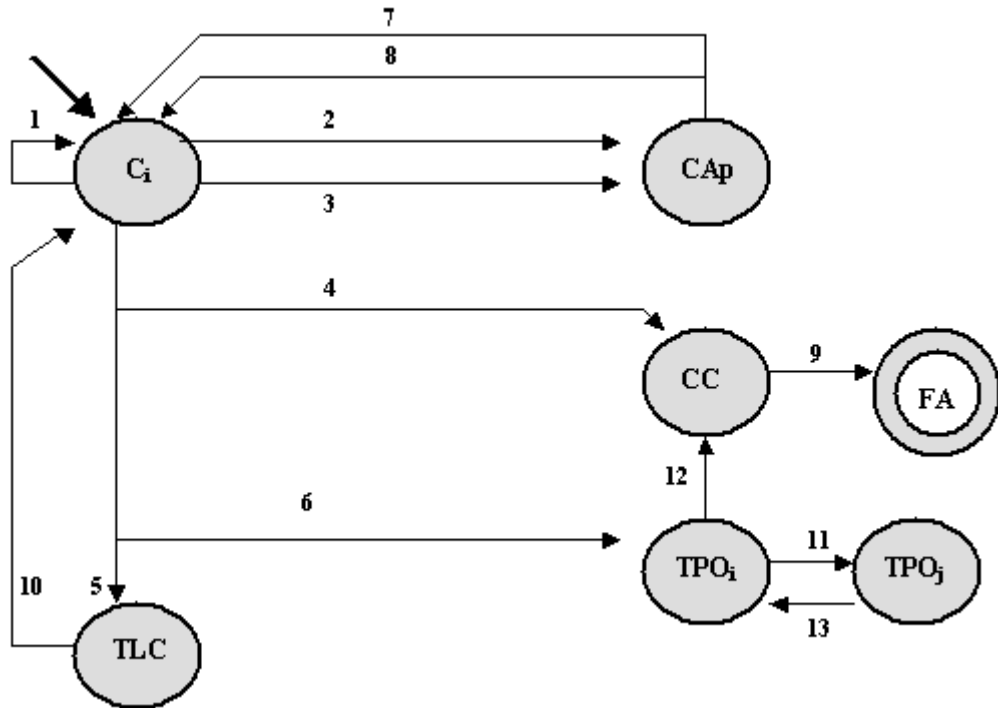


Figura 4.7: Autômato representando a fase de ensino.

O autômato de pilha é: o conjunto de estados é: $Q = \{C_i, TLC, TPO_i, TPO_j, CAP, CC, FA\}$. O alfabeto de entrada é: $\Sigma = \{(termos), (exemplos), (lemas), (conjecturas), (hipótese), (princípios), (opinião)\}$. O alfabeto da pilha é: $\Gamma = \{ \$, (termos), (exemplos), (lemas), (conjecturas), (hipótese), (princípios), (opinião)\}$. O estado inicial é: $q_0 = \{C_i\}$. O conjunto de estados de aceitação: $F = \{FA\}$. Por fim, a função de transição é:

- $\delta_1: (C_i, \epsilon, \epsilon) \rightarrow (C_i, \$)$
- $\delta_2: (C_i, (termos), \$) \rightarrow (CAP, (termos))$
- $\delta_3: (C_i, (exemplos), (termos)) \rightarrow (CAP, (exemplos))$
- $\delta_4: (C_i, (hipótese), (conjecturas)) \rightarrow (CC, (hipótese))$
- $\delta_5: (C_i, (conjecturas), (lemas)) \rightarrow (TCL, (conjecturas))$
- $\delta_6: (C_i, (princípios), \$) \rightarrow (TPO_i, (princípios))$
- $\delta_7: (CAP, \epsilon, (termos)) \rightarrow (C_i, (termos))$
- $\delta_8: (CAP, (lemas), (exemplos)) \rightarrow (C_i, (lemas))$
- $\delta_9: (CC, \epsilon, (hipótese)) \rightarrow (FA, \epsilon)$
- $\delta_{10}: (TCL, \epsilon, (conjecturas)) \rightarrow (C_i, (conjecturas))$
- $\delta_{11}: (TPO_i, \epsilon, (princípios)) \rightarrow (TPO_j, (princípios))$
- $\delta_{12}: (TPO_i, (hipótese), (opinião)) \rightarrow (CC, (hipótese))$
- $\delta_{13}: (TPO_j, (opinião), (princípios)) \rightarrow (TPO_i, (opinião))$

A linguagem reconhecida pelo autômato da figura 4.7 é: $x = (termos) (exemplos) (lemas) (conjecturas)$, $y = (princípio) (opinião)$ e $z = (hipótese)$, então, $L_{FE} = \{xz \text{ ou } yz\}$. Significa que o instrutor (estado C_i) envia os *termos* e os *exemplos* da amostra de aprendizagem. No componente de aprendizagem (estado CAP) é extraído um conjunto de *lemas*,

que formam um conjunto de *conjecturas*. Uma destas conjecturas é escolhida como a *hipótese* a ser testada, que é armazenada no componente de conhecimento (estado CC). Este é o caso do ensino indireto. No ensino direto, o instrutor, a partir de um conjunto de *princípios*, gera uma *opinião*. Esta opinião é a hipótese a ser testada. Começa a fase de avaliação (estado FA).

4.3.2.2 Fase de avaliação

O autômato de pilha referente fase de avaliação do ciclo de modelagem é mostrado na figura 4.8.

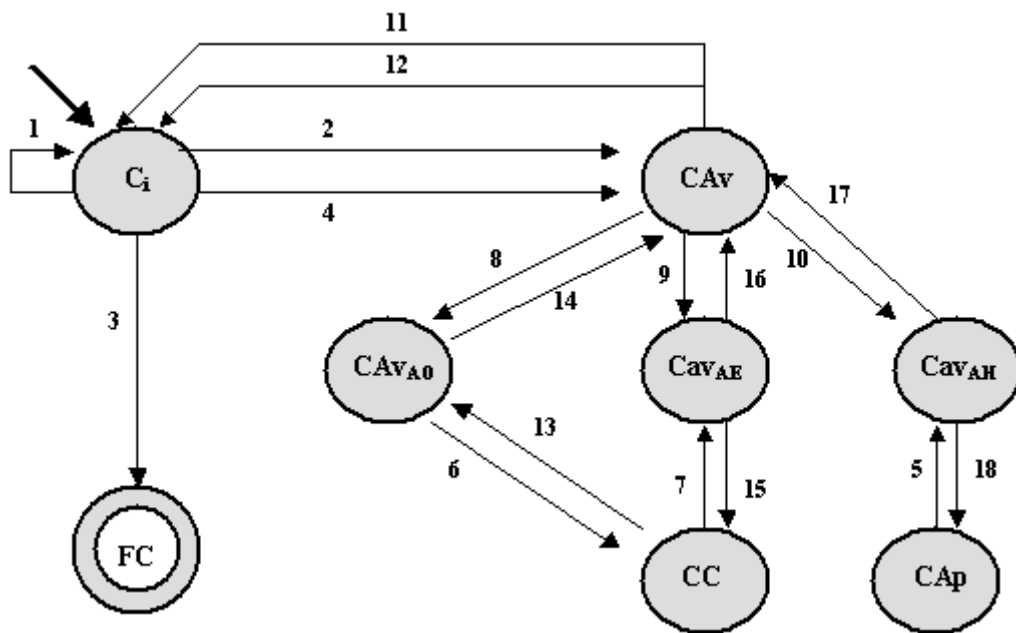


Figura 4.8: Autômato representando a fase de avaliação

O autômato de pilha é uma 6-tupla: o conjunto de estados é: $Q = \{C_i, CA_p, CC, CA_v, CAV_{AO}, CAV_{AE}, CAV_{AH}, FC\}$. O alfabeto de entrada é: $\Sigma = \{(amostra_{av}), (adeqont), (adeqeps), (adeqheu), (amostra_{ap}), (adequações)\}$. O alfabeto da pilha é: $\Gamma = \{A1, A2, A3, \&, \$, (amostra_{av}), (adequações)\}$. O estado inicial é: $q_0 = \{C_i\}$. O conjunto de estados de aceitação: $F = \{FC\}$. Por fim, a função de transição é:

- $\delta_1: (C_i, \varepsilon, \varepsilon) \rightarrow (C_i, \$)$
- $\delta_2: (C_i, (amostra_{av}), (termos)) \rightarrow (CA_v, (amostra_{av}))$
- $\delta_3: (C_i, \varepsilon, (adequações)) \rightarrow (FC, \varepsilon)$
- $\delta_4: (C_i, (termos), \$) \rightarrow (CA_v, (termos))$
- $\delta_5: (CA_p, (amostra_{ap}), A3) \rightarrow (CA_{vAH}, A3)$
- $\delta_6: (CC, (hipótese), A1) \rightarrow (CA_{vAO}, A1)$
- $\delta_7: (CC, (hipótese), A2) \rightarrow (CA_{vAE}, A2)$
- $\delta_8: (CA_v, \varepsilon, (amostra_{av})) \rightarrow (CA_{vAO}, A1)$
- $\delta_9: (CA_v, \varepsilon, A2) \rightarrow (CA_{vAE}, A2)$

- $\delta_{10}: (CAV, \epsilon, A3) \rightarrow (CAV_{AH}, A3)$
- $\delta_{11}: (CAV, (adequa\c{c}oes), \&)) \rightarrow (Ci, (adequa\c{c}oes))$
- $\delta_{12}: (CAV_i, \epsilon, (termos)) \rightarrow (Ci, (termos))$
- $\delta_{13}: (CAV_{AO}, \epsilon, A1) \rightarrow (CC, A1)$
- $\delta_{14}: (CAV_{AO}, (adeqont), A1) \rightarrow (CAV, A2)$
- $\delta_{15}: (CAV_{AE}, \epsilon, A2) \rightarrow (CC, A2)$
- $\delta_{16}: (CAV_{AE}, (adeqeps), A2) \rightarrow (CAV, A3)$
- $\delta_{17}: (CAV_{AH}, (adeqheu), A3) \rightarrow (CAV, \&)$
- $\delta_{18}: (CAV_{AH}, \epsilon, A3) \rightarrow (CAp, A3)$

A linguagem reconhecida pelo autômato da figura 4.8 é: $x = (termos) (amostra_{av})$ (hipótese) (adeqont) (hipótese) (adeqeps), $y = (amostra_{ap})$, $z = (adeqheu)$ e $w = (adequa\c{c}oes)$, então, $L_{FA} = \{xyzw \text{ ou } xzw\}$. Significa que o instrutor (estado Ci) envia os *termos* e a amostra de avaliação (*amostra_{av}*) ao componente de avaliação (CAV). A hipótese corrente, armazenada no componente de conhecimento (estado CC) é consultada para se fazer a adequação ontológica (*adeqont*), e depois novamente a hipótese é consultada para se fazer a adequação epistêmica (*adeqeps*). Quando o ensino é indireto, a amostra de aprendizagem (*amostra_{ap}*) é consultada, no componente de aprendizagem (estado CAp), para se fazer a adequação heurística (*adeqheu*). As *adequa\c{c}oes* são enviadas ao instrutor. Começa a fase de correção (estado FC).

4.3.2.3 Fase de correção

O autômato de pilha que ilustra a fase de correção do ciclo de modelagem é mostrado na figura 4.9.

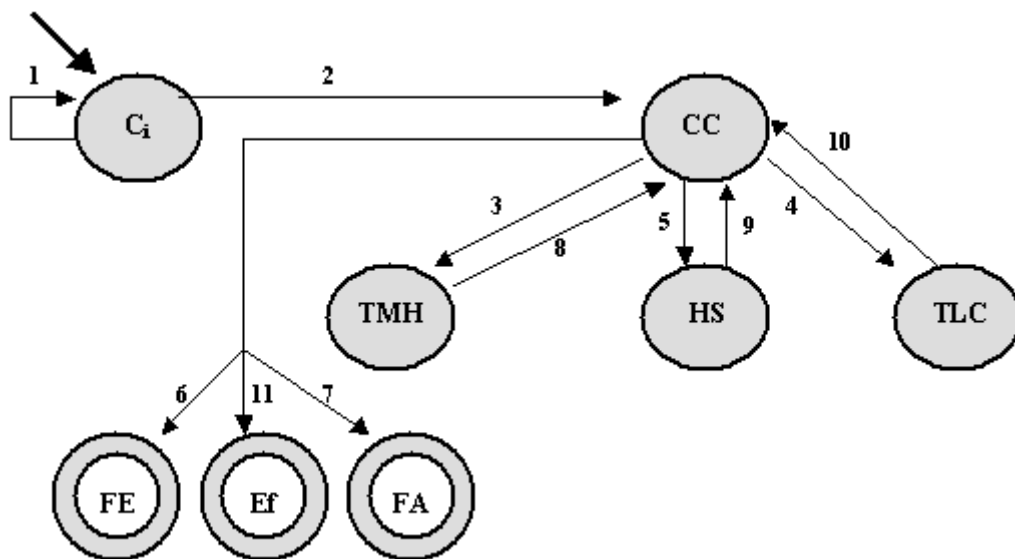


Figura 4.9: Autômato representando a fase de correção.

O autômato de pilha é uma 6-tupla: o conjunto de estados é: $Q = \{Ci, CC, TMH, TLC, HS, FE, FA\}$. O alfabeto de entrada é: $\Sigma = \{(críticas), (hip_pre_mod), (hip_não_ad), (am_te_me), (hip_sat), (hipótese)\}$. O alfabeto da pilha é: $\Gamma = \{\&, \$, (críticas), (hipótese)\}$. O estado inicial é: $q_0 = \{Ci\}$. O conjunto de estados de aceitação: $F = \{FE, FA\}$. Por fim, a função de transição é:

$$\begin{aligned} \delta_1: (Ci, \epsilon, \epsilon) &\rightarrow (Ci, \$) \\ \delta_2: (Ci, (críticas), \$) &\rightarrow (CC, (críticas)) \\ \delta_3: (CC, (hip_pre_mod), (críticas)) &\rightarrow (TMH, \&) \\ \delta_4: (CC, (hip_nao_ad), (críticas)) &\rightarrow (TLC, \&) \\ \delta_5: (CC, (hip_sat), (críticas)) &\rightarrow (HS, \&) \\ \delta_6: (CC, (am_te_me), (críticas)) &\rightarrow (FE, \epsilon) \\ \delta_7: (CC, \epsilon, (hipótese)) &\rightarrow (FA, \epsilon) \\ \delta_8: (TMH, (hipótese), \&) &\rightarrow (CC, (hipótese)) \\ \delta_9: (HS, (hipótese), \&) &\rightarrow (CC, (hipótese)) \\ \delta_{10}: (TLC, (hipótese), \&) &\rightarrow (CC, (hipótese)) \\ \delta_{11}: (CC, \epsilon, (críticas)) &\rightarrow (Ef, \epsilon) \end{aligned}$$

A linguagem reconhecida pelo autômato da figura 4.8 é: $x = (críticas)$, $y_1 = (hip_pré_mod)$, $y_2 = (hip_não_ad)$, $y_3 = (hip_sat)$, $y_4 = (am_te_me)$ e $z = (hipótese)$, então, $L_{FC} = \{xy_1z \text{ ou } xy_2z \text{ ou } xy_3z \text{ ou } xy_4\}$. Significa que as *críticas* do instrutor pode ser: hipótese precisa de modificações (*hip_pré_mod*), a hipótese não é adequada (*hip_não_ad*), os termos e/ou a amostra foram mal escolhidos (*am_te_me*) ou que a hipótese é satisfatória (*hip_sat*). Após a *hipótese* ser criticada, pode-se retornar a fase de avaliação (estado FA). Quando os termos ou a amostra de aprendizagem são considerados mal escolhidos, retorna-se a fase de ensino (estado FE), indicando que o ciclo deve ser refeito. O instrutor também pode decidir encerrar o ciclo de modelagem (estado Ef).

O uso de autômato de pilha (do inglês *pushdown automaton*) como linguagem de modelagem deve-se ao fato de que pretende-se mostrar que a interação instrutor-agente é simples, ao ponto de não precisar de máquina de Turing, ou seja, precisar de um modelo computacional mais poderoso, que reconheça as linguagens que representam esta interação. Todas as linguagens aqui mostradas são livres do contexto - são reconhecidas por autômato de pilha [Sipser 97]. Um ponto importante a ser observado é que as linguagens reconhecidas pelos autômatos de pilha considerados, além de serem livres do contexto, também são regulares, ou seja, são reconhecidas por autômatos finitos. O motivo pelo qual usou-se um modelo computacional mais poderoso é que, usando autômatos de pilha, ganha-se mais em expressividade.

Outras ferramentas de modelagem e especificação poderiam ter sido utilizadas, como G-CPN, usada para a especificação formal do agente SAID. Devido a sua simplicidade, o autômato de pilha foi escolhido como a ferramenta mais adequada para a vertente formal deste trabalho.

Vamos a vertente experimental.

4.3.3 Vertente experimental

Para analisar o ciclo de modelagem segundo a vertente experimental, é usado um domínio de aplicação simples, que serve como ilustração. O domínio escolhido foi o *mun-do dos arcos*, devido a sua simplicidade e por ter sido usado em outros trabalhos [Winston 75]. Considere que o especialista do domínio, ou instrutor, dispõe da seguinte amostra de exemplos, que ele decide usar como amostra de aprendizagem (figura 4.10). Este domínio foi também usado como uma aplicação ilustrativa pelo WebContract + Praal (WCP), o programa demonstrador da metodologia do φ -calculus (anexo A).

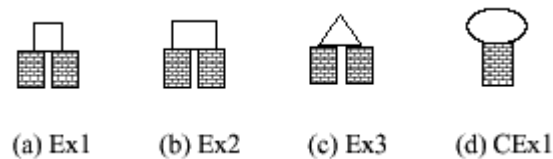


Figura 4.10: Amostra de aprendizagem.

O instrutor, a partir de sua própria experiência e discernimento, extraiu os seguintes termos da amostra (figura 4.11), como critério a ser usado para a inclusão dos exemplos, tanto na fase de ensino, como na fase de avaliação. Estes termos são apresentados aqui numa forma hierárquica.

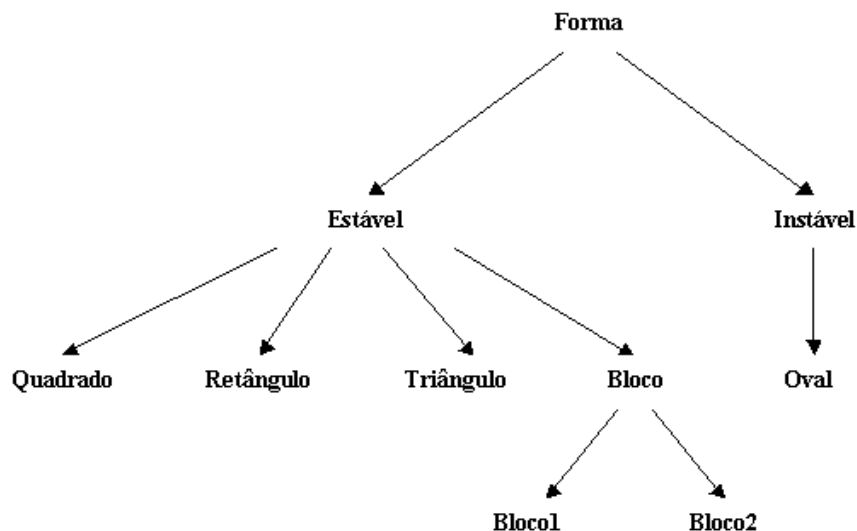


Figura 4.11: Termos escolhidos pelo instrutor.

O ciclo de modelagem começa.

4.3.3.1 Fase de ensino

Digamos que o ensino é indireto. No início da fase de ensino, o instrutor apresenta a seu agente os termos a serem considerados. Logo em seguida, vem a inclusão dos exemplos, que inclui dois mecanismos: *formulação* e *exemplificação*. O mecanismo de formulação representa as *observações* do instrutor em cada exemplo, enquanto mecanismo de *exemplificação* refere-se a classificação dos exemplos em positivos ou negativos. Os exemplos são apresentados ao agente um por um. Por exemplo, para a inclusão do objeto da figura 4.10(a), a fórmula do exemplo é: quadrado+ e bloco1+ e bloco2+ (o símbolo "+" é para indicar que estes termos foram verificados no objeto), e a exemplificação é: arco+ (pois o objeto constitui-se em um exemplo positivo do conceito *arco*, ou seja, é considerado um arco). Da mesma forma, os outros objetos da amostra são incluídos: retângulo+ e bloco1+ e bloco2+ e arco+ (figura 4.10(b)), triângulo+ e bloco1+ e bloco2+ e arco+ (figura 4.10(c)), oval+ e bloco1+ e arco- (figura 4.10(d), aqui o objeto não é considerado como uma instância do conceito *arco*).

Da perspectiva do instrutor, o objetivo do processo de aprendizagem do agente é extrair da coleção dos exemplos da amostra um número de regularidades que podem ser combinadas para compor os *lemas*. Digamos que os lemas extraídos dos objetos são:

- (1) triângulo+ \rightarrow bloco2+, arco+;
- (2) retângulo+ \rightarrow bloco2+, arco+;
- (3) quadrado+ \rightarrow bloco2+, arco+;
- (4) arco- \rightarrow oval+;
- (5) oval+ \rightarrow arco-;
- (6) bloco2+ \rightarrow arco+;
- (7) arco+ \rightarrow bloco2+;
- (8) bloco2+ \rightarrow bloco1+;

A partir destes lemas, o instrutor os organiza em *conjecturas*, neste caso, digamos que o instrutor resolveu organizar os lemas em duas conjecturas: conjectura 1 (bloco2+ \rightarrow arco+; arco+ \rightarrow bloco2+; bloco2+ \rightarrow bloco1+; arco- \rightarrow oval+ e oval+ \rightarrow arco-) e conjectura2 (bloco2+ \rightarrow arco+; arco+ \rightarrow bloco2+ e bloco2+ \rightarrow bloco1+). O instrutor escolhe, como a *hipótese* a ser testada como verificando o conceito *arco*, a conjectura 1. Desta forma, os lemas da conjectura passam a formar os *postulados* da hipótese.

Para verificar se a hipótese escolhida realmente representa o conceito *arco*, que o

instrutor deseja obter, começa a fase de avaliação.

4.3.3.2 Fase de avaliação

Como dito anteriormente, o objetivo do processo de avaliação é decidir se o conhecimento aprendido pelo agente corresponde ao conhecimento que o instrutor tentou passar para ele durante a fase de ensino. Considere o objeto da figura 4.12 como o objeto constituinte da amostra de avaliação (esta amostra é composta por apenas um exemplo).



Figura 4.12: Amostra de avaliação.

Com base nos termos mostrados na figura 4.11, a amostra é inserida para ser avaliada (quadrado+ e triângulo+ e bloco1+ e bloco2+ e arco-).

A avaliação é feita em três níveis: *ontológico*, *epistêmico* e *heurístico*. Dada uma fórmula F , produzida através do mecanismo de formulação, e a hipótese H , sendo manipulada, a questão a ser avaliada no nível ontológico é: todos os termos que aparecem em F aparecem também na descrição intencional (nos postulados) da hipótese H ? Esta pergunta é importante para que o instrutor avalie se os termos definidos por ele são ou não adequados. Se a resposta for *sim*, então a adequação ontológica é verificada; caso contrário, diz-se que houve uma inadequação ontológica. Para o nível epistêmico, a pergunta é: para cada termo t que aparece em F , e para cada postulado Pt descrevendo H , se t aparece em Pt , a avaliação obedece ao que é fortalecido em Pt ? Esta pergunta é importante para o instrutor avaliar a qualidade/pertinência dos postulados da hipótese corrente sendo testada. Se a resposta é *sim*, adequação epistêmica, se a resposta é *não*, inadequação epistêmica. Nível heurístico: se F formula um exemplo positivo (negativo) do conceito e Sf , que é apenas a fórmula sem a sua classificação em instância ou não do conceito, aparece no mínimo em um dos exemplos da descrição extensional do conceito, o(s) exemplo(s) é(são) positivo (negativo)? Por isso que o agente deve ter acesso a amostra de aprendizagem para fazer a avaliação à nível heurístico. Se a resposta é *sim*, adequação heurística, se a resposta é *não*, inadequação heurística. No caso deste nível de adequação, o agente pode não ter a resposta – ele não consegue encontrar na amostra de aprendizagem nenhum exemplo que possa ajudá-lo respondê-la. Quando isto acontece, o agente fica em silêncio. Neste trabalho, este acontecimento é chamado de *problema do silêncio*.

A resposta do processo de avaliação para a amostra de avaliação é:

- **nível ontológico.** A resposta é *não*, pois os termos *quadrado* e *triângulo* não são definidos nos postulados da hipótese.
- **nível epistêmico.** A resposta também é *não*, pois viola os postulados da hipótese ($\text{bloco2+} \rightarrow \text{arco+}$; $\text{arco+} \rightarrow \text{bloco2+}$).
- **nível heurístico.** A resposta é *silêncio*, pois nenhum exemplo da amostra de aprendizagem contém uma formulação semelhante ao exemplo da amostra de avaliação.

Como dito anteriormente, o resultado vindo da fase de avaliação serve apenas como orientação a seu instrutor, pois cabe a este decidir o que fazer com a hipótese sendo testada. De posse do resultado da amostra de avaliação da figura 4.12, o instrutor pára para pensar.

4.3.3.3 Fase de correção

Ao pensar acerca do resultado proveniente da fase de avaliação, o instrutor julgou que os postulados $\text{bloco2+} \rightarrow \text{arco+}$ e $\text{arco+} \rightarrow \text{bloco2+}$ são desnecessários, e resolveu retirá-los da hipótese – esta hipótese modificada constitui-se em uma *objeção*, e suas partes componentes de *provas*. Os postulados da hipótese agora são: $\text{bloco2+} \rightarrow \text{bloco1+}$; $\text{arco-} \rightarrow \text{oval+}$ e $\text{oval+} \rightarrow \text{arco-}$.

Após esta decisão, o instrutor achou que não deveria retornar a fase de avaliação, mas sim recomeçar um novo ciclo para aperfeiçoar a hipótese sendo testada. Um ciclo só pode ser seguido por um outro ciclo em que o ensino seja direto, então, desta forma, retorna-se a fase de ensino.

4.3.3.4 Considerações

Começando um novo ciclo, o ensino agora é direto. Analisando a hipótese corrente, o instrutor chegou a conclusão de que deveria "dizer" a seu agente que um arco não possui duas formas sobre os blocos (como o objeto da amostra de avaliação). Então, ele acrescenta um a *opinião* sobre isto, cujo conjunto (unitário neste caso) de *princípios* é: *exclusão_mútua*(quadrado+ e retângulo+ e triângulo+ e oval+). O princípio anterior é incorporado ao conjunto de postulados da hipótese: $\text{bloco2+} \rightarrow \text{bloco1+}$; $\text{arco-} \rightarrow \text{oval+}$; $\text{oval+} \rightarrow \text{arco-}$ e *exclusão_mútua*(quadrado+ e retângulo+ e triângulo+ e oval+). Começa a

fase de avaliação.

O instrutor resolveu usar uma outra amostra de avaliação (figura 4.13). Este objeto é apresentado ao agente (quadrado+ e bloco1+ e bloco2+ e arco-).

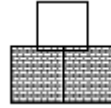


Figura 4.13: Nova amostra de avaliação.

A resposta da avaliação, nos três níveis de adequação, para esta amostra é:

- **nível ontológico.** A resposta é *sim*, pois os termos *quadrado*, *bloco1* e *bloco2* são definidos nos postulados da hipótese.
- **nível epistêmico.** A resposta é *sim*, pois não viola os postulados da hipótese.
- **nível heurístico.** A resposta é *não*, pois há um exemplo da amostra de aprendizagem (figura 4.10(a)) que contém uma formulação semelhante ao exemplo da amostra de avaliação. O exemplo da amostra de aprendizagem é positivo e o da amostra de avaliação é negativo.

Neste momento o instrutor está muito confuso. Depois de refletir bastante, ele percebeu que na hierarquia de termos não foi considerada a distância entre os dois blocos, que evitariam a inadequação heurística no caso de objetos como o da amostra de avaliação. Sendo assim, o instrutor decidiu refazer a hierarquia de termos (figura 4.14) e refazer todo o processo de modelagem.

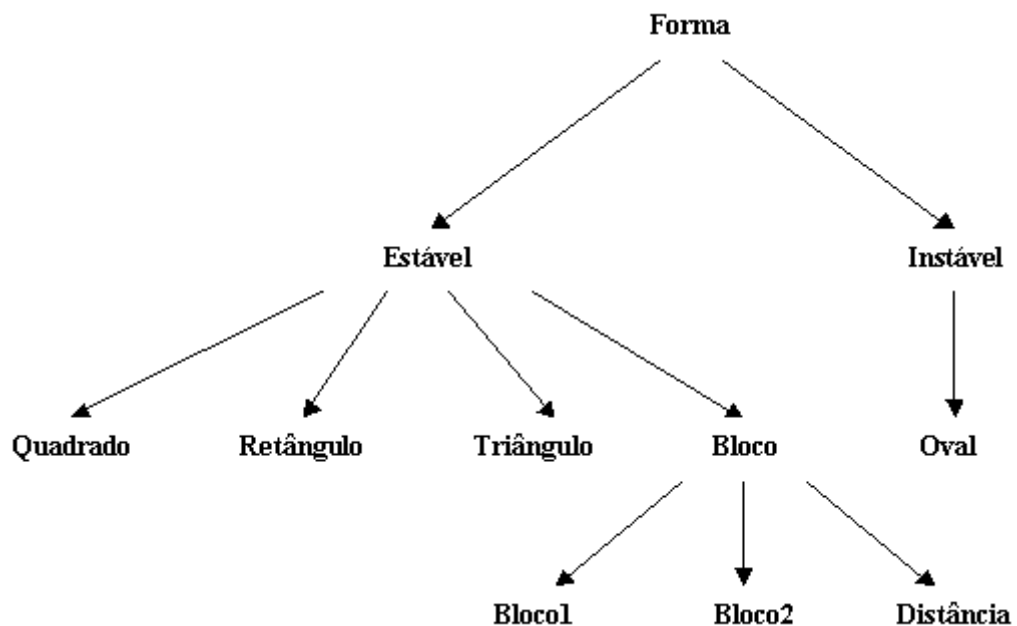


Figura 4.14: Nova hierarquia de termos.

4.4 Discussão

Do ponto de vista de agentes inteligentes, a aprendizagem provê autonomia no sentido de que o comportamento do agente é determinado por sua própria experiência. Segundo Russell & Norvig [Russell 95], um agente aprendiz deve possuir:

- **Componente de execução.** Responsável por selecionar uma ação a ser realizada.
- **Componente de aprendizagem.** Responsável por incrementar o conhecimento do agente, recebendo conhecimento do Componente de Execução e um retorno do Crítico indicando a pertinência das ações do agente.
- **Crítico.** O papel desse componente é informar ao Componente de Aprendizagem sobre o desempenho do agente, através de uma medida de desempenho.
- **Gerador de problemas.** Responsável por sugerir ações que confrontem o agente com novas experiências.

Estes componentes são mostrados na figura 4.15. A aprendizagem pode ser vista como o resultado da interação entre o agente e o mundo, e como resultado da observação do processo de tomada de decisão do agente.

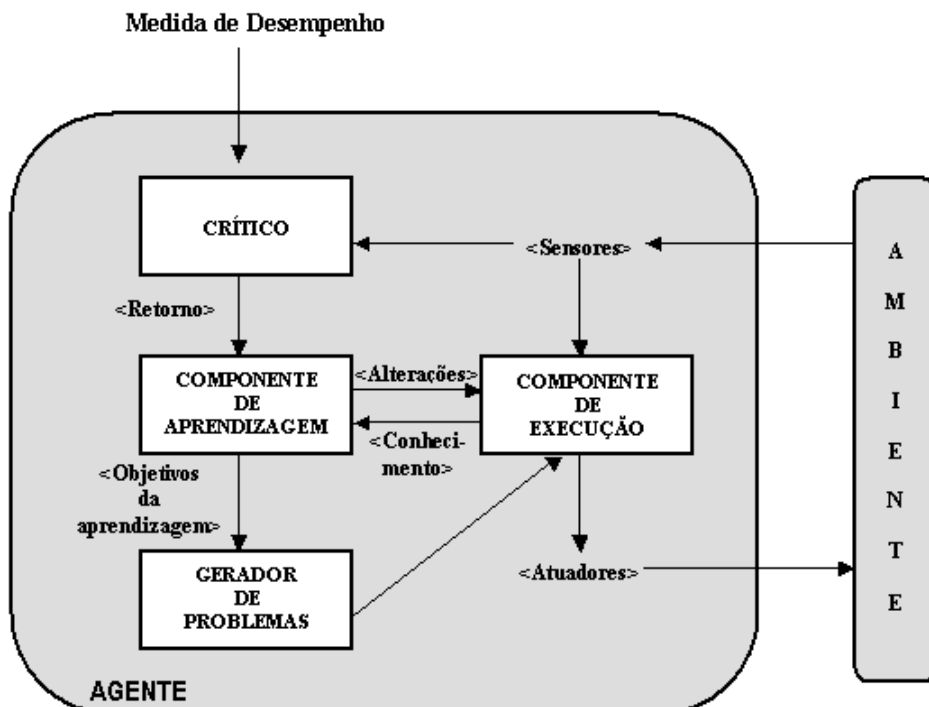


Figura 4.15: Modelo genérico de agente aprendiz proposto por Russell e Norvig.

O SAID [Nóbrega 98] pode ser visto como um refinamento do modelo genérico de agente aprendiz de Russell. Seus componentes podem ser descritos sucintamente como:

- **Componente de execução.** Responsável por aplicar a conjectura a um objeto enviado pelo cliente e obter uma crença, que será retornada ao cliente.
- **Componente de aprendizagem.** Responsável por gerar uma conjectura (pelos mecanismos de abdução e indução), a partir da amostra obtida do oráculo e fazer evoluir esta conjectura, a partir das críticas recebidas do mestre.
- **Crítico (mestre).** Responsável pela avaliação do desempenho do agente através do envio de críticas. No caso do SAID, o crítico não julga diretamente a ação realizada pelo agente, mas sua justificativa. Dessa forma, além da crença enviada ao agente, o aprendiz é capaz de produzir também uma explicação.
- **Gerador de problemas (sonda).** Sugere objetos associados a crenças para o Componente de Execução para que produza uma crença, a confronto com aquela recebida e argumente.

A figura 4.16 ilustra esta adaptação.

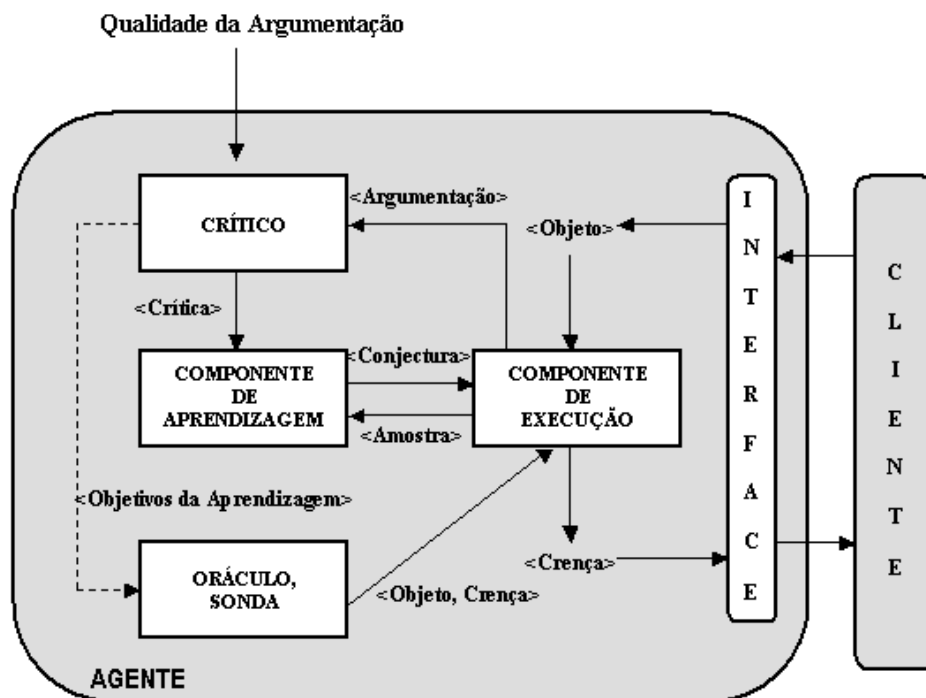


Figura 4.16: Agente SAID adaptado ao modelo genérico de aprendiz de Russell e Norvig.

O modelo genérico de agente aprendiz baseado em raciocínio assíncrono interativo, proposto neste trabalho, também pode ser adaptado ao modelo de Russell e Norvig (figura 4.17). Seus componentes são:

- **Componente de Execução (Interface).** Responsável por intermediar a interação en-

tre o agente aprendiz e seu instrutor. O especialista do domínio apresenta uma amostra de exemplos, ou diretamente o conhecimento que ele deseja modelar, e espera que o sistema devolva o conceito acerca do domínio. Quando o Componente de Aprendizagem devolve os lemas extraídos da amostra de exemplos, é a partir da Interface que o instrutor os organiza em conjecturas.

- **Componente de Aprendizagem.** Responsável por gerar um conjunto de lemas a partir da amostra obtida através do instrutor.
- **Crítico (Componente de Conhecimento).** Responsável por armazenar a hipótese sendo testada. Após a avaliação, a hipótese recebe críticas do especialista do domínio – como dito anteriormente, a intuição do instrutor é usada como medida de desempenho.
- **Gerador de problemas (Componente de avaliação).** Responsável por testar a hipótese corrente. A partir de uma amostra composta de exemplos positivos e negativos, o Componente de Avaliação retorna um resultado estimado de acordo com três níveis de adequação: ontológica, epistêmica e heurística. Para a avaliação, a hipótese corrente é requisitada.

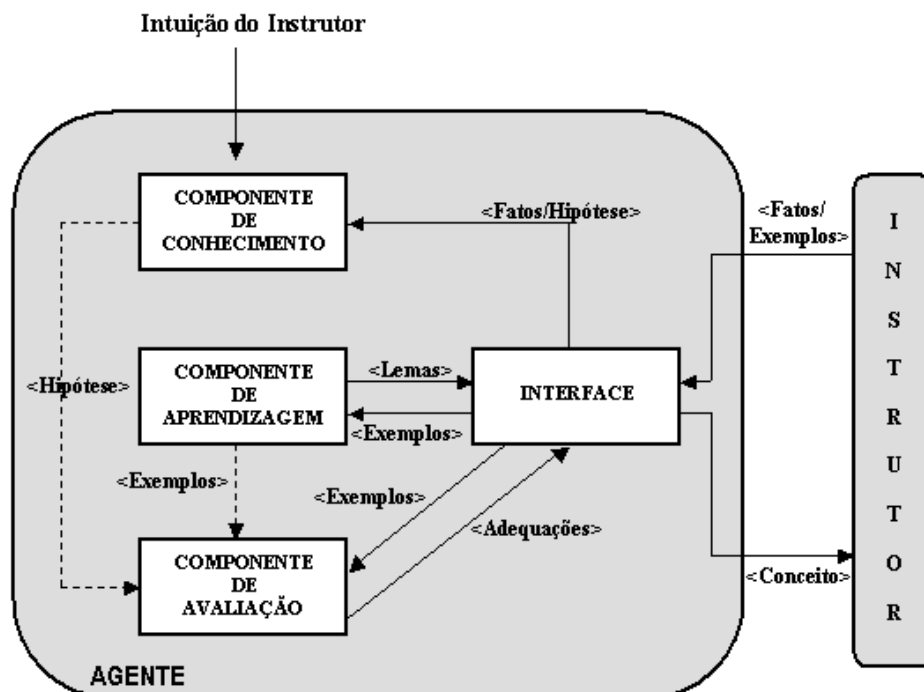


Figura 4.17: Agente baseado em raciocínio assíncrono interativo adaptado ao modelo genérico de agente aprendiz de Russell.

4.5 Conclusão

Neste trabalho foi apresentado um modelo genérico de agente aprendiz baseado em raciocínio assíncrono interativo. Este agente interage com seu instrutor (humano). A interação instrutor-aprendiz é vista segundo um ciclo/processo de aprendizagem dividido em três fases: ensino, avaliação e correção. A hipótese corrente, sendo testada, é julgada segundo três níveis de adequação: ontológico, epistêmico e heurístico. Este novo trabalho é um dos esforços do GIA na direção da concepção e desenvolvimento de sistemas de apoio à descoberta científica.

No próximo capítulo são apresentadas as considerações e perspectivas acerca do trabalho.

Capítulo 5

Conclusão

5.1 Considerações finais

Um ser humano com especialidade em algum domínio de conhecimento deseja testar a veracidade de alguma informação que armazena com a ajuda de um colaborador. Ele, o especialista do domínio, observa os *fatos* que existem espontaneamente na natureza e procura estabelecer *leis* que representem relações entre fatos. O seu objetivo é construir um *conceito* que sintetize exatamente o que deseja expressar, testando várias *hipóteses* . Estas hipóteses são estruturadas pelo colaborador na forma de *postulados* , que servem de suporte para elas.

O especialista do domínio poderia apresentar a seu colaborador, que é dotado de alguma capacidade de aprendizagem, uma amostra de *exemplos* que representam instâncias do conceito que ele deseja obter. A partir desta amostra, seu colaborador construiria *lemas* que seriam, posteriormente, agrupados em *conjecturas* , e uma destas conjecturas seria a hipótese escolhida como aquela a ser testada. Uma outra maneira do especialista obter o conceito é emitir uma *opinião* acerca do que ela pensa sobre o domínio em foco. Ao apresentar a seu colaborador esta opinião, ele deve suportá-la dizendo quais os *princípios* que o levam a pensar desta forma.

Depois de expor a seu ajudante o seu ponto de vista acerca do domínio, o especialista poderia observar o comportamento de seu colaborador, confrontando-o com uma nova amostra de exemplos. Se o especialista julgar que o comportamento não está em *adequação* com o que ele pensa, pode demonstrar uma *objeção* modificando a hipótese que está sendo testada. Os termos que sustentam esta objeção seriam as *provas* .

A situação ilustrada acima pode ser reconhecida como um processo de modelagem de um conhecimento. O especialista do domínio desempenha o papel de um instrutor, pois

transmite algum tipo de informação, e seu colaborador o de um agente artificial, pois comporta-se como uma entidade de software/hardware de teste para seu instrutor. φ -calculus é a ferramenta teórica que busca formalizar a interação entre agente humano instrutor e agente artificial aprendiz/colaborador. Este trabalho de dissertação buscou conceber um modelo genérico de agente aprendiz baseado nas atitudes mentais matemáticas de φ -calculus e na idéia de raciocínio assíncrono interativo, constituindo-se em mais um dos esforços do Grupo de Inteligência Artificial (GIA) para a concepção de sistemas de apoio à descoberta científica. As vertentes complementares aqui mostradas (conceitual, formal e experimental) serviram para visualizar a interação instrutor-agente aprendiz a partir de diferentes focos.

5.2 Trabalhos Futuros

Como propostas para trabalhos futuros, é sugerido:

- Como foi exposto na fase de ensino da vertente conceitual, um ciclo de modelagem (seja com ensino direto ou indireto) só pode ser seguido por um outro ciclo cujo ensino é direto. É necessário uma melhor avaliação desta restrição, pois seria interessante que o agente possuísse meios de avisar a seu instrutor que existem exemplos inconsistentes na amostra de aprendizagem, ou seja, exemplos com formulações iguais e exemplificações diferentes. Com este problema resolvido, os ciclos de modelagem seguintes poderiam ter ensino indireto.
- Revisar φ -calculus completamente finalizado e fazer um paralelo entre ele e este trabalho. Na qualidade de uma ferramenta teórica ainda não completamente finalizada, este trabalho baseou-se no que é definido/conhecido acerca de φ -calculus. Omissões e equívocos podem naturalmente surgir. Apesar disto, este trabalho constitui-se numa contribuição às idéias de φ -calculus, principalmente no que diz respeito a algumas definições/combinções das atitudes mentais matemáticas.
- Revisar o processo de busca por adequação, principalmente a adequação heurística. Este ponto é motivo de muita discussão entre os idealizadores da ferramenta e ainda não existe um consenso acerca deste processo. O que foi considerado dentro deste trabalho diz respeito ao ponto de vista de um dos pesquisadores, que resolveu basear-se nos três tipos de adequação de McCarthy e Hayes para uma representação de conhecimento.

- Procurar resolver o *problema do silêncio* – o problema do silêncio é detectado quando o agente aprendiz não possui uma resposta para o seu instrutor com relação ao julgamento no nível de adequação heurística. Este problema, verificado também no projeto SAID, é um dos mais difíceis e se constitui em um grande desafio, tanto porque não se tem um consenso acerca do processo de busca heurística..
- Após terminados os trabalhos citados anteriormente, uma ferramenta computacional poderia ser construída. É importante que o problema da legibilidade dos lemas seja considerado, pois o instrutor terá acesso direto a eles. Também a interface usada para a inserção de princípios e composição de opiniões deve ser a mais simples possível.
- Aplicar este modelo em domínios de problemas mais complexos - nas áreas de ensino à distância, comércio eletrônico, etc.

O trabalho foi proposto no âmbito do projeto Capes/Cofecub (de cooperação interuniversitária entre o Brasil e a França) e surgiu da negociação entre o Grupo de Inteligência Artificial (GIA), do lado brasileiro, e a equipe de Jean Sallantin (Laboratoire d'Informatique, Robotique et Microélectronique de Montpellier, França), do lado francês. A doutoranda Germana Menezes da Nóbrega, estudante suportada pela CAPES em Montpellier, é orientada por Jean Sallantin (portanto integrante da equipe deste) e principal colaboradora deste trabalho de dissertação. A ferramenta teórica ϕ -calculus é o tema de tese de doutorado da referida estudante.

Como continuação destes trabalhos, é proposto um tema de tese de doutorado (*Aprendizagem e Interação para a formação de teorias científicas: aplicação à análise de seqüências genéticas*) cujo objetivo é determinar os âmbitos de raciocínio para a concepção e a supervisão por operadores humanos da formação de teorias científicas. Pretende-se aplicá-lo à anotação de seqüências genéticas.

Referências bibliográficas

- [Bacon 84] **F. Bacon**, *Novum Organum*, coleção Os Pensadores, 3 Edição, Editora Abril, São Paulo (SP), 1984 (citado em [Nóbrega 98]).
- [Bittencourt 96] **G. Bittencourt**, *Inteligência Artificial: Ferramentas e Teorias*, 10ª Escola de Computação, Campinas, 8 a 13 de julho de 1996.
- [Carvalho 99] **J. N. Carvalho**, *S3O: Um Método de Busca de Similaridades em Objetos Estruturados*, dissertação de mestrado em Informática, COPIN/CCT/UFPB, Campina Grande, 1999.
- [Carvalho 01] **J. N. Carvalho, E. Ferneda, R. Gheyi e P. L. Dantas**, *Um Sistema Multi-Agentes de Apoio à Descoberta: Elementos de Construção de Hipóteses*, ENIA'2001.
- [Castro 98] **E. Castro, G. Gachelin e J. Sallantin**, *M@int Technical Report Convention N 972930132*, LIRMM, 1998.
- [Cheeseman 88] **P. Cheeseman, D. Freeman, J. Kelly, M. Self, J. Stutz e W. Taylor**, *AUTOCLASS: a bayesian classification system*, Proceedings of Fifth International Conference on Machine Learning (pp. 54-64), An Arbor, MI: Morgan Kaufmann, 1988.
- [Fajtlowicz 88] **S. Fajtlowicz**, *On conjectures of GRAFFITI*, Discrete Mathematics, 72, 113-118, 1988.
- [Feigenbaum 71] **E. A. Feigenbaum, B. G. Buchanan e J. Lederberg**, *On generality and problem solving: a case study using the DENDRAL program*, In Machine Learning (vol. 6), Edinburg: Edinburg University Press, 1971.
- [Ferber 99] **J. Ferber**, *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*, Addison-Wesley, 1999.
- [Ferneda 92a] **E. Ferneda, M. Py, P. Reitz e J. Sallantin**, *L'agent rationnel SAID: une application en géométrie*, Proceedings of the First European Colloquium on Cognitive Science, pp. 175 – 192, Orsay (França), 1992.
- [Ferneda 92b] **E. Ferneda**, *Conception d'un agent rationnel et examen de son raisonnement en géométrie*, tese de doutorado, Université de Montpellier II, Montpellier (França), 1992.
- [FIDAL 00] *Procédé et système de conception interactive d'une base de connaissance libre d'ambigüités ainsi que l'outil informatique pour la mise en oeuvre du procédé et du système*, FIDAL/CNRS, patente requerida em

21 de novembro de 2000 (nº0014999).

- [Ginsberg 90] **M. L. Ginsberg**, *Bilattices and modal operators*, Journal of Logic and Computacion, nº 1, 1990.
- [Gueyi 00] **R. Gueyi, R. A. Dixit, G. M. da Nóbrega e E. Ferneda**, *Desenvolvimento de um Sistema de Apoio à Descoberta*, Anais do Seminário'2000, Salvador (Ba), 2000.
- [Gudwin 99] **R. Gudwin**, *Introdução à Teoria dos Agentes*, <http://www.dca.fee.unicamp.br/~gudwin/IA365G/>, 1999.
- [Haton 90] **J. Haton e M. Haton**, *L'intelligence Artificielle*, Coleção *Que sai-je?*, Press Universitaires de France, Paris, junho de 1990.
- [Jong 97] **H. de Jong e A. Rip**, *The Computer Revolution in Science: Steps Towards the Realization of Computer-supported Discovery Environments*, Artificial Intelligence, Volume 91, Number 2, abril de 1997.
- [Lakatos 76] **L. Lakatos**, *A Lógica do Descobrimento Matemático: Provas e Refutações*, Zahar, Rio de Janeiro (RJ), 1976.
- [Langley 81] **P. Langley, G. L. Bradshaw e H. A. Simon**, *BACON5: the discovery of conservation laws*, Proceedings IJCAI-81, 1981.
- [Langley 92] **P. Langley et al**, *Scientific Discovery Computational Explorations of the Creative Process*, MIT Press, 1992.
- [Langley 96] **P. Langley**, *Elements of Machine Learning*, Morgan Kauffmann Publishers Inc., 1996.
- [Langley 98] **P. Langley**, *The Computer-Aided Discovery of Scientific Knowledge*, <http://www.isle.org/~langley/papers/discovery.ds98.ps.gz>, 1998.
- [Langley 00] **P. Langley**, *The Computational Support of Scientific Discovery*, <http://www.isle.org/~langley/papers/discovery.ijhcs99.ps.gz>, 2000.
- [Ledesma 97] **L. Ledesma, A. Pérez, D. Borrajo e L. M. Laita**, *A Computational Approach to George Boole's Discovery of Mathematical Logic*, Artificial Intelligence, volume 91, Number 2, abril de 1997.
- [Lee 98] **Y. Lee, B. G. Buchanan e J. M. Aronis**, *Knowledge-based learning in exploratory science: learning rules to predict rodent carcinogenicity*, Machine Learning, 30, 217-240, 1998.
- [Lenat 83a] **D. B. Lenat**, *The Role of Heuristics in Learning by Discovery: The Three Case Studies*, In: J. G. Carbonell, R. S. Michalski and T. M. Mitchell (Eds), *Machine Learning: An Artificial Intelligence Approach*, Morgan Kaufmann, Palo Alto (EUA),1983 (citado em [Nóbrega 98]).
- [Lenat 83b] **D. B. Lenat**, *Theory Formation by Heuristic Search – The Nature of Heuristics II: Background and Examples*, Artificial Intelligence, 1983.
- [Liquière 90] **M. Liquière e J. Sallantin**, *INNE (INduction in NETworks): a structural learning algorithm for noisy examples*, Proceedings of the Fourth European Working Session on Learning (Montpellier – 1989), Pitman • Morgan Kaufmann Publishers Inc., Londres (Inglaterra), 1990.
- [Martins 98] **L. F. Martins, E. Ferneda e J. N. Carvalho**, *S3O: Um Método de*

Busca de Similaridades em Objetos Estruturados, Anais do Seminário'98, Salvador (Ba), 1998.

- [Mephu 93] **E. Mephu N'guifo**, *Prediction of Primate Splice Junction Gene Sequences With a Cooperative Knowledge Acquisition System*, Proceedings of the 1st International Conference on Intelligent Systems for Molecular Biology, Washington (EUA), Eds: L Hunter, D. Searls, J. Shavlik, AAAI/MIT Press, 1993.
- [Michalski 83] **R. S. Michalski, J. G. Carbonell e T. M. Mitchell**, *Machine Learning: An Artificial Intelligence Approach*, Morgan Kaufmann Publishers Inc., 1983.
- [Müller 87] **J. P. Müller**, *Contribution à l'étude d'un agent rationnel: spécification en logique intensionnelle et implantation*, tese de doutorado, Institut National Polytechnique de Grenoble, Grenoble (França) 1987 (citado em [Nóbrega 98]).
- [Newell 62] **A. Newell, J. C. Shaw e H. A. Simon**, *The Process of Creative Thinking*, In: H. E. Gruber et al (Eds), *Contemporary Approaches to Creative Thinking*, Nova York (EUA), 1962 (citado em [Nóbrega 98]).
- [Nóbrega 98] **G. M. da Nóbrega**, *Especificação Formal de um Sistema de Apoio à Descoberta Científica*, dissertação de mestrado em Informática, COPIN/CCT/UFPB, Campina Grande (Pb), 1998.
- [Nóbrega 00] **G. M. da Nóbrega, E. Castro, P. Malbos, J. Sallantin e S. Cerri**, *A framework for supervised conceptualizing*, Proceedings of the ECAI-00 Workshop on Applications of Ontologies and Problem-Solving Methods, Berlin (Alemanha), 2000.
- [Nóbrega 01a] **G. M. da Nóbrega, J. Sallantin e P. Malbos**, *Modelling through human-computer interactions and mathematical discourse*, Model-Based Reasoning: Scientific Discovery, Technological Innovation, Values – MBR'01, Pavia (Itália), 2001.
- [Nóbrega 01b] **G. M da Nóbrega, S. A. Cerri, e J. Sallantin**, *DIAL: serendipitous DIAlectic Learning*, In T. Okamoto, R. Hartley, Kinshuk, and J. P. Klus, editors, IEEE International Conference on Advanced Learning Technologies - ICALT'2001, pages 109-110, Madison, Wisconsin (EUA), 2001.
- [Nóbrega 02a] **G. M da Nóbrega, P. Malbos, e J. Sallantin**, *Modeling Through Human-Computer Interactions and Mathematical Discourse*. In L. Magnani, N. J. Nersessian, and C. Pizzi, editors, *Logical and computational aspects of model-based reasoning*, Applied Logic Series. Kluwer Academic Press, 2002.
- [Nóbrega 02b] **G. M da Nóbrega, S. A. Cerri e J. Sallantin**, *On the Social Rational Mirror: Learning E-commerce in Web-served Learning Environment*. In S. A. Cerri, G. Gouardères, and F. P. Costa, editors, *Intelligent Tutoring Systems 6th International Conference – ITS 2002*, Biarritz (France), 5 a 8 de junho de 2002.
- [Popper 93] **K Popper**, *A Lógica da Pesquisa Científica*, Cultrix, São Paulo (SP), 1998.
- [Reitz 92] **P. Reitz**, *Contribution a L'étude des Environnements*

- D'apprentissage: Spécification et Prototypage*, tese de doutorado, LIRMM, Université de Sciences et Techniques du Languedoc – Montpellier II, Montpellier (França), 1992 (citado em [Nóbrega 98]).
- [Russell 95] **S. Russell e P. Norvig**, *Artificial Intelligence: A Modern Approach*, Prentice-Hall, 1995.
- [Sallantin 91a] **J. Sallantin, J. Quinqueton, C. Barboux e J. –P. Aubert**, *Théories Semi-Empiriques: Éléments de Formalisation*, Revue d'Intelligence Artificielle, 5(1), Paris (França), 1991.
- [Sallantin 91b] **J. Sallantin, J. –J. Szczeciniarz, C. Barboux, M. –S. Lagrange e M. Renaud**, *Théories semi-empiriques: conceptualisation et illustrations*, Revue d'Intelligence Artificielle, 5(1), Paris (França), 1991.
- [Sallantin 97] **J. Sallantin**, *Les Agents Intelligents: essai sut la rationalité des calculs*, Editions Hermes, França, 1997 (citado em [Silva 98]).
- [Sallantin 00] **J. Sallantin, G. M. da Nóbrega**, *Les temps, la pensée et le calcul: une stratification des claculs pour fonder une cosmologie*, Épistémologiques philosophie, sciences, histiore – Revue internationale, Paris/São Paulo, 2000.
- [Silva 98] **M. E. de Souza e Silva**, *De Agentes Rationais a Agentes Semióticos: um Estudo sobre a Aplicação da Semiótica na Concepção de Sistemas Inteligentes*, dissertação de mestrado em Informática, COPIN/CCT/UFPB, Campina Grande (Pb), 1998.
- [Simon 97] **H. A. Simon, R. E. Valdés-Pérez e D. H. Sleeman**, *Scientific Discovery and Simplicity of Method*, Artificial Intelligence, Volume 91, Number 2, abril de 1997.
- [Sipser 97] **M. Sipser**, *Introduction to the Theory of Computation*, PWS Publishing Company, 1997.
- [Valdés-Pérez 95] **R. E. Valdés-Pérez**, *Machine Learning in chemistry: new results*, Artificial Intelligence, 74, 191-201, 1995.
- [Valdés-Pérez 96] **R. E. Valdés-Pérez**, *Computer Science Research on Scientific Discovery*, <http://www-2.cs.cmu.edu/afs/cs/user/valdes/Mosaic/Postscript/ker96.ps>, 1996.
- [Valdés-Pérez 99] **R. E. Valdés-Pérez**, *Principles of Human Computer Collaboration for Knowledge Discovery in Science*, <http://www-2.cs.cmu.edu/~sci-disc/Abstracts/aij99.html>, 1990.
- [Villareal 89] **M. J. Villareal-Fernandez**, *Construction par apprentissage de modèles d'objects complexes*, tese de doutorado, Université de Montpellier II, Montpellier (França), abril de 1989 (citado em [Nóbrega 98]).
- [Vuillemin 84] **J. Vuillemin**, *Nécessité Ou Contingence, L'aporie de Diodore et Les Systèmes Philosophiques*, Les Editions de Minuit, Paris (França), 1984 (citado em [Nóbrega 01]).
- [Winston 75] **P. H. Winston**, *Learning Structural Descriptions from Examples, the Psychology of Computer Vision*, New York, McGraw Hill, 1975.

Anexo A

WCP - um programa demonstrador da metodologia empregada em ϕ -calculus

WebContract é um propagador de restrições [Castro 98]. Praal (Propositional gRAph And Learning) é um algoritmo de aprendizagem a partir de exemplos. O sistema WebContract + Praal (WCP) reúne estas duas ferramentas para servir como um demonstrador para o ϕ -calculus.

Utilizando o *mundo dos blocos* como ilustração, três exemplos e um contra-exemplo (figura A.1), a construção do conceito arco, pelo agente aprendiz WCP, é descrita a seguir [Nóbrega 01b].

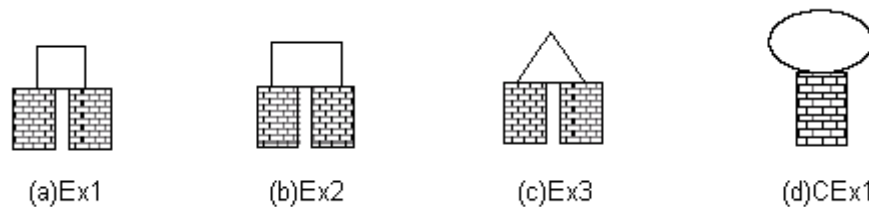


Figura A.1: Três exemplos e um contra-exemplo do conceito arco.

Antes de introduzir os exemplos no sistema, o usuário deve primeiro organizar, sob uma forma hierárquica, os termos que designam os objetos que constituem os exemplos (figura A.2). É obtida uma estrutura de árvore que contém (a) *termos* como seus nós e (b) relações *é-um* como suas ligações. No WCP, o usuário deve construir a hierarquia de termos de acordo com a sintaxe típica do sistema – o conteúdo é composto por um nome e pela própria hierarquia.

Agora, assumindo que tal hierarquia contém todos os termos necessários para representar os exemplos, o usuário pode então introduzi-los no sistema (figura A.3). Isto será feito indicando na tela que termos estão presentes/ausentes, com a finalidade de apreender o que é observado pelo usuário em cada exemplo. O WCP automaticamente assumirá co-

mo presentes todos os termos que são ancestrais de um termo presente no exemplo (propagação de presença). Da mesma forma, o WCP assumirá como ausentes todos os termos que são descendentes de um termo ausente do exemplo (propagação de ausência).

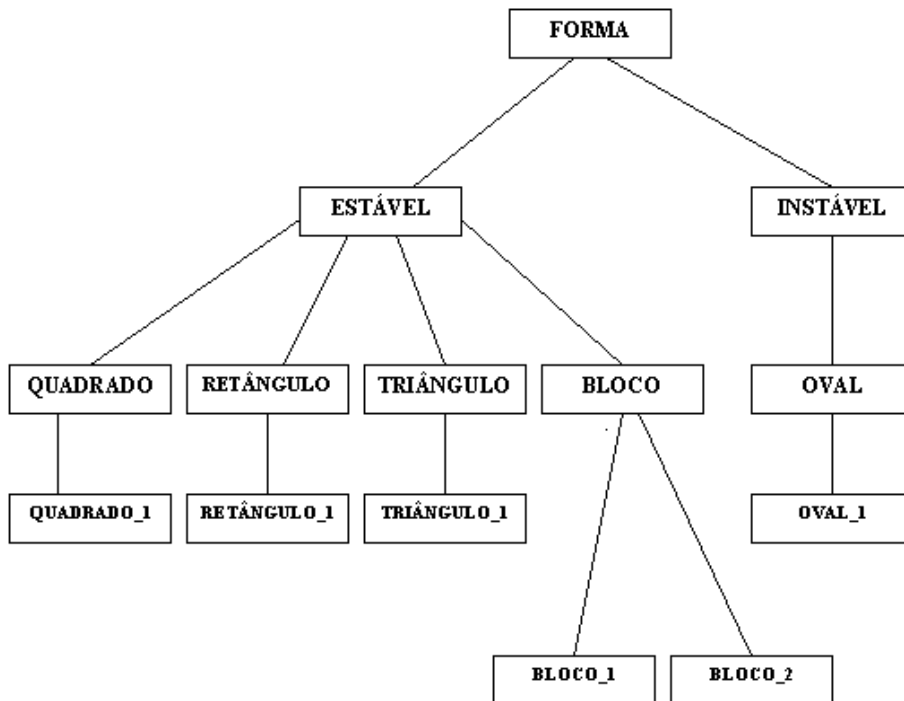


Figura A.2: Mundo dos blocos: hierarquia de conceitos.

Uma vez que o usuário definiu a presença/ausência de todos os termos representando as formas observadas em um exemplo, interações adicionais são requeridas para introduzir o exemplo atual:

- Primeiro, o usuário deve escolher um termo da hierarquia para anexar o exemplo;
- Depois, para introduzir o exemplo, o usuário deve pressionar o botão “validar”. O exemplo é então adicionado à coleção (se existir) de exemplos anexados anteriormente ao termo (figura A.4).

Depois de ter definido todos os exemplos, o usuário pode disparar o processo de aprendizagem (figura A.5). O objetivo do processo de aprendizagem é extrair da coleção de exemplos um número de regularidades que compõem regras que:

- Irão compor a teoria pretendida;
- Poderão ser usadas pelo usuário para incrementar a teoria, bem como restringir o uso de termos.

Restrição	Explicação	Fórmula lógica
com $t_1 t_2 \dots t_n; s_1 s_2 \dots s_n$	Comando	$t_1 \wedge t_2 \wedge \dots \wedge t_n \rightarrow s_1 \vee s_2 \vee \dots \vee s_n$
exc $t_1 t_2 \dots t_n$	Exclusão	$\neg t_1 \vee \neg t_2 \vee \dots \vee \neg t_n$
ex2 $t_1 t_2 \dots t_n$	Exclusão múltipla	$(\neg t_1 \vee t_2 \vee \dots \vee t_n) \wedge (t_1 \vee \neg t_2 \vee \dots \vee t_n) \wedge \dots \wedge (t_1 \vee t_2 \vee \dots \vee \neg t_n)$
ver $t_1 t_2 \dots t_n$	Verificação	$t_1 \vee t_2 \vee \dots \vee t_n$

Tabela A.1: Tipos de restrições manipuladas pelo WCP.

Uma vez que o usuário “chamou” o processo de aprendizagem, o sistema propõe um número de regras lógicas compondo as regularidades encontradas na coleção de exemplos. As regras serão submetidas ao julgamento do usuário. Para disponibilizar as regras lógicas ao usuário, WCP deve traduzir cada uma delas em um dos *tipos de restrições* que ele é capaz de manipular (tabela A.1). Cada regra traduzida é submetida ao julgamento do usuário antes de serem consideradas pelo sistema. Logo após serem analisadas, o usuário pressiona o botão “validar” para o sistema reconhecê-las.

Uma vez que a construção de exemplos é restringida pela adição de restrições, chegou o momento de validá-las. Tal validação pode ser desenvolvida pela proposição de um número de exemplos desconhecidos e pela observação da reação do sistema – se o sistema age como esperado. Quando um exemplo (positivo ou negativo) é construído, uma *violação de restrição* pode ser detectada pelo sistema (figura A.6).

Depois que uma contradição foi detectada, pode-se remover as restrições violadas para evitar situações contraditórias (figura A.7). Depois de removidas, o exemplo (positivo ou negativo) pode ser finalmente introduzido sem qualquer violação.



Figura A.3: Introdução do primeiro exemplo da amostra.

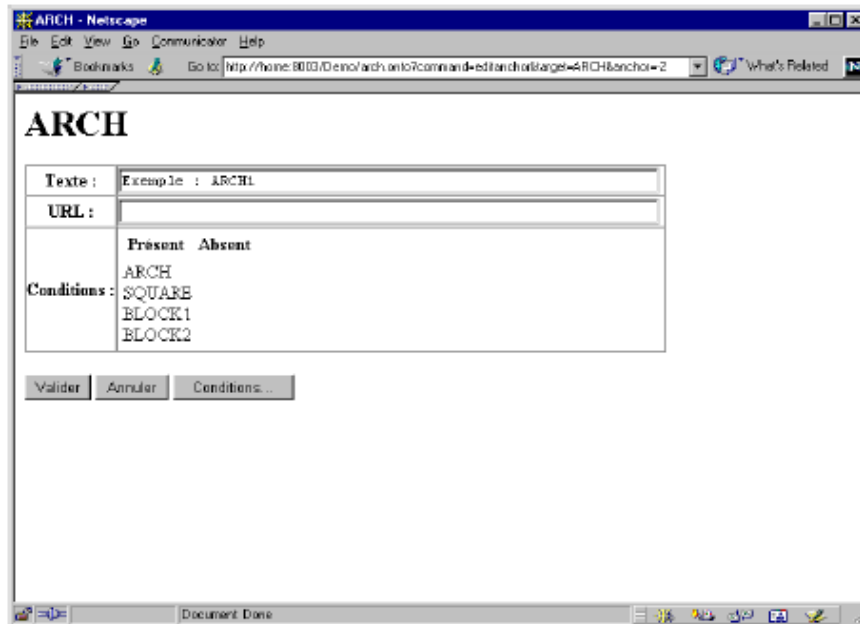


Figura A.4: Adição do exemplo à coleção de exemplos anexados anteriormente.

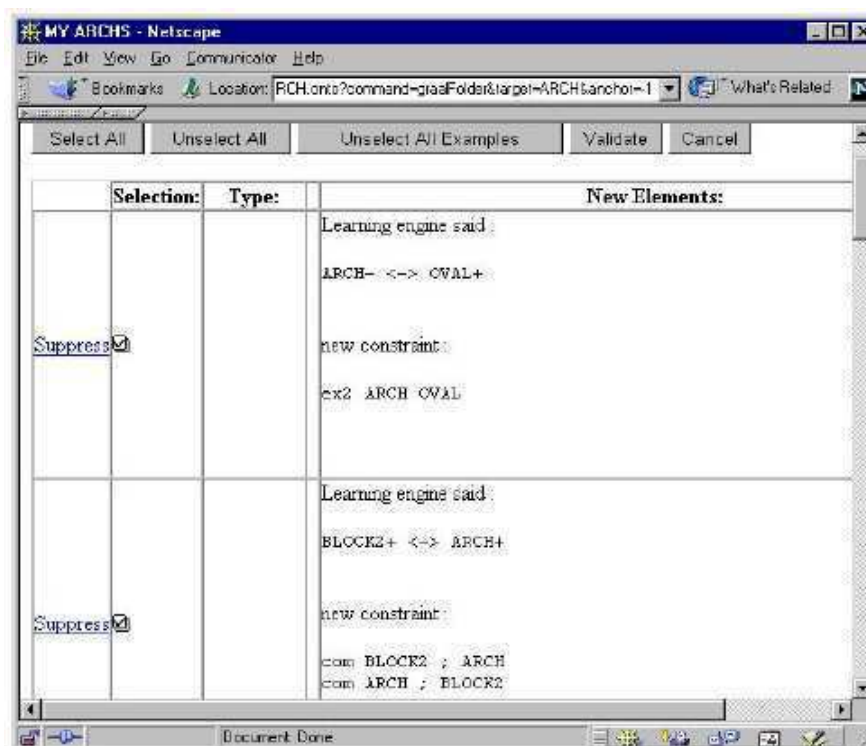


Figura A.5: O processo de aprendizagem

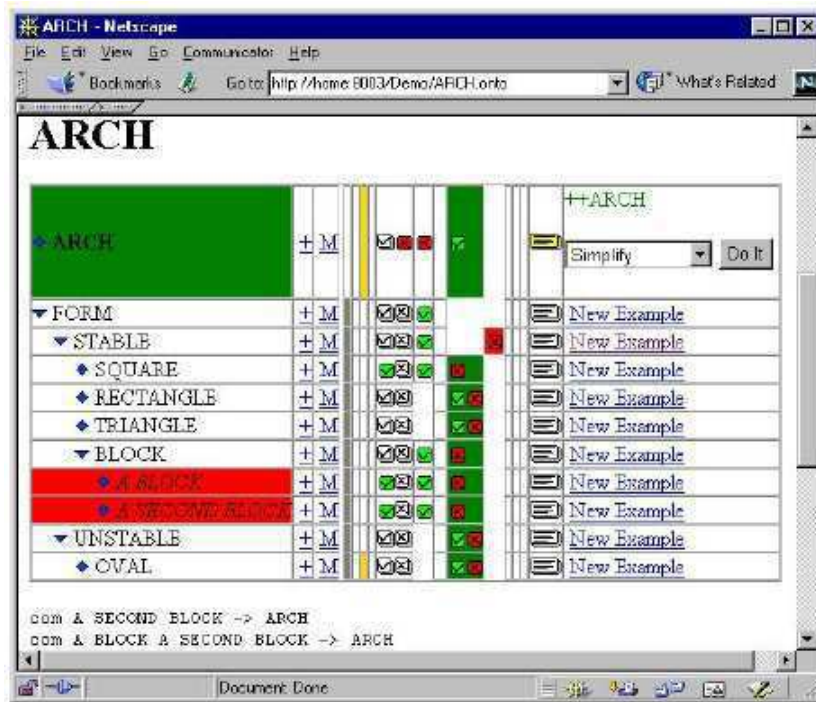


Figura A.6: Uma violação de restrição

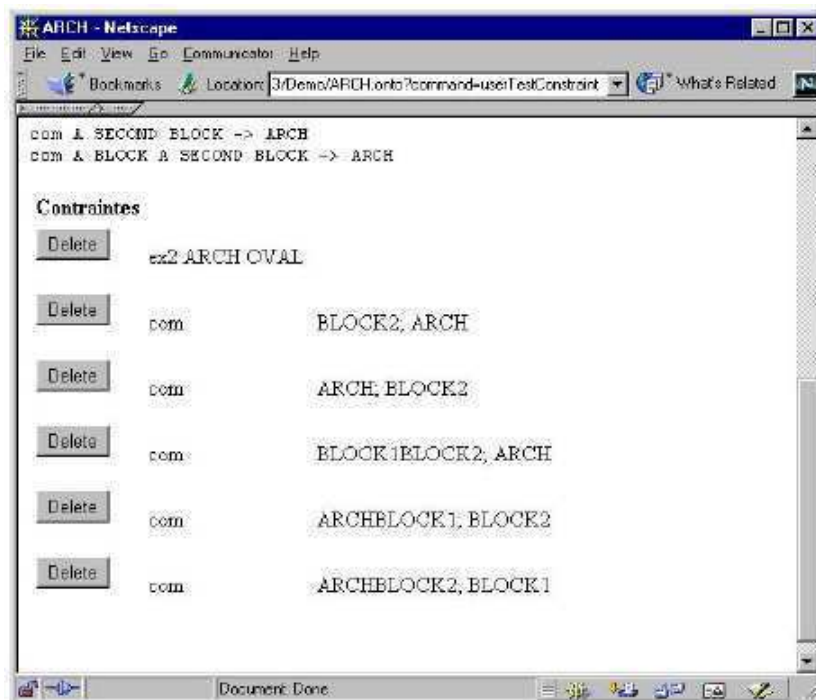


Figura A.7: Remoção das restrições violadas.

Anexo B

Autômato Pushdown

O *autômato pushdown* é como um autômato finito não determinístico, mas tem um novo componente chamado *pilha*. A pilha fornece memória adicional além da capacidade do autômato finito. A pilha permite que o autômato pushdown reconheça algumas linguagens não regulares [Sipser 96].

Um autômato pushdown (APD) pode gravar símbolos na pilha e lê-los mais tarde. A gravação de um símbolo “empilha” os outros símbolos na pilha. A qualquer momento o símbolo do topo da pilha pode ser lido e removido. Os símbolos restantes então sobem uma posição. A gravação de um símbolo é chamada comumente de *empilhamento* daquele símbolo e a remoção é chamada *desempilhamento*. Note que os acessos à pilha, sejam para leitura ou gravação só podem ser realizados no topo. Em outras palavras, a pilha é uma estrutura do tipo LIFO (last in, first out). Se uma informação está na pilha e é gravado outro campo, a primeira torna-se inacessível até que a segunda seja removida.

Pratos servem como exemplo de uma pilha. Um prato está no topo até que seja posto um outro sobre este, que move-se para baixo. Uma pilha de um autômato pushdown é como uma pilha de pratos, com cada prato tendo um símbolo gravado. A pilha é importante porque ela pode conter um número infinito de símbolos.

A definição formal de um autômato pushdown é similar à de um autômato finito, exceto pela pilha. A pilha é uma estrutura que contém símbolos de algum alfabeto. A máquina pode usar alfabetos diferentes para a entrada e a pilha, portanto agora especificamos os alfabetos Σ (da entrada) e Γ (da pilha).

A parte mais importante da definição formal de um autômato é a função de transição, já que esta descreve seu comportamento. Lembre que $\Sigma_\epsilon = \Sigma \cup \{\epsilon\}$ e $\Gamma_\epsilon = \Gamma \cup \{\epsilon\}$. O domínio da função de transição é $Q \times \Sigma_\epsilon \times \Gamma_\epsilon$. Assim, o estado atual, o símbolo lido

da pilha e da cadeia de entrada determinam o próximo movimento do autômato pushdown. Os símbolos podem ser ϵ , o que faz a máquina mover sem gastar símbolo algum da pilha ou da entrada.

Para a imagem da função de transição, precisamos considerar o que o autômato pode fazer em situações específicas. Ele pode entrar em um novo estado e possivelmente escrever um símbolo na pilha. A função δ pode indicar essa ação retornando um símbolo de Q junto a um símbolo de Γ_ϵ , isto é, um símbolo de $Q \times \Gamma_\epsilon$. Devido ao fato de que permitimos não determinismo nesse modelo, uma situação pode ter vários movimentos válidos para um determinado grupo de símbolos lidos. A função de transição incorpora o não determinismo da forma usual, retornando um conjunto de membros de $Q \times \Gamma_\epsilon$, isto é, um membro de $P(Q \times \Gamma_\epsilon)$. Unindo, a nossa função de transição δ toma a forma $\delta: Q \times \Sigma_\epsilon \times \Gamma_\epsilon \rightarrow P(Q \times \Gamma_\epsilon)$.

Um *autômato pushdown* é uma 6-tupla $(Q, \Sigma, \Gamma, \delta, q_0, F)$, onde Q, Σ, Γ e F são conjuntos finitos, e

1. Q é o conjunto de estados,
2. Σ é o alfabeto de entrada,
3. Γ é o alfabeto da pilha,
4. $\delta: Q \times \Sigma_\epsilon \times \Gamma_\epsilon \rightarrow P(Q \times \Gamma_\epsilon)$ é a função de transição,
5. $q_0 \in Q$ é o estado inicial, e
6. $F \subseteq Q$ é o conjunto de estados finais.

Um autômato pushdown $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$ computa como a seguir. Ele aceita a entrada w se w pode ser escrita como $w = w_1 w_2 \dots w_m$, onde cada $w_i \in \Sigma_\epsilon$ e a seqüência dos estados $r_1, r_2, \dots, r_m \in Q$ e as cadeias $s_1, s_2, \dots, s_m \in \Gamma^*$ existe de forma que satisfaçam as três seguintes condições. As cadeias s_i representam a seqüência dos símbolos da pilha que M tem no estado final da computação.

1. $r_0 = q_0$ e $s_0 = \epsilon$. Esta condição quer dizer que M começa propriamente no estado inicial e com a pilha vazia.
2. Para $i = 0, \dots, m-1$, temos $(r_{i+1}, b) \in \delta(r_i, w_{i+1}, a)$, onde $s_i = at$ e $s_{i+1} = bt$, para algum a e $b \in \Gamma_\epsilon$ e $t \in \Gamma^*$. Essa condição atesta que M move-se propriamente de acordo com o estado, com a pilha e o símbolo lido da entrada.
3. $r_m \in F$. Essa condição atesta que um estado final ocorre no fim de uma entrada.