UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO
MESTRADO EM CIÊNCIA DA COMPUTAÇÃO

Um Sistema para Aprendizagem de Demonstrações Dedutivas em Geometria Euclidiana

por

MÁRCIA RODRIGUES NOTARE

Dissertação submetida à avaliação,
como requisito parcial para a obtenção do grau de Mestre
em Ciência da Computação

Prof. Dr. Tiarajú Asmuz Diverio
Orientador

Porto Alegre, setembro de 2001.

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Notare, Márcia Rodrigues

Um Sistema para Aprendizagem de Demonstrações Dedutivas em Geometria Euclidiana / por Márcia Rodrigues Notare. – Porto Alegre: PPGC da UFRGS, 2001.

124 p. : il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR – RS, 2001. Orientador: Diverio, Tiarajú Asmuz.

1. Informática na Educação. 2. Matemática da Computação. 3. Demonstração de Teoremas. I. Diverio, Tiarajú Asmuz. II. Título.

Agradecimentos

Em primeiro lugar, agradeço à minha família, em especial aos meus pais Edison e Enilsa, que foram os maiores responsáveis pelo incentivo ao estudo e pesquisa. Obrigada pela confiança, paciência e pelo apoio financeiro.

Gostaria de agradecer ao meu orientador, Prof. Tiarajú Asmuz Diverio, que ajudou a definir e solidificar as idéias, que muitas vezes pareciam puras divagações, sem forma ou fundamento. Que infinitamente me incentivou e apoiou, confiando no meu trabalho e na minha competência. Que se mostrou severo quando necessário, mas também amigo e compreensivo nos momentos certos.

Agradeço a Profa. Maria Alice Gravina, minha eterna orientadora, primeira pessoa a confiar no meu trabalho, ainda na graduação em Matemática, e que me encaminhou para o mundo da pesquisa. Obrigada pela super força que sempre me deste e pelas idéias que ajudaram o solidificar este trabalho.

Agradeço também ao meu companheiro de todas as horas, Edgar, pelo apoio e paciência em todos os momentos que dediquei à concretização deste trabalho.

Agradeço ainda aos colegas que, de uma forma ou outra, ajudaram-me nesta caminhada, trocando idéias e acompanhando o desenvolvimento do trabalho e àqueles que foram grandes companheiros de festas e momentos agradáveis. Em especial, agradeço ao bolsista Guilherme Drehmer, pela sua colaboração na fase de implementação do sistema, ao colega Júlio Henrique, que muito me ajudou na fase de concepção e ao Prof. Paulo Blauth Menezes, com os quais compartilhei trabalhos e artigos.

A todos vocês, muito obrigada.

Sumário

Lista de Abreviaturas

A1	Axioma 1
A3	Axioma 3
C1	Círculo 1
C2	Círculo 2
CASYS	International Conference on Computing Anticipatory Systems
D15	Definição 15
D20	Definição 20
Eurocast	International Conference on Computer Aided Systems Theory and Technology
fig.	Figura
IA	Inteligência Artificial
LEEG	Learning Environment on Euclidean Geometry
MOSCA	Mestre + Oráculo + Sonda + Cliente + Aprendiz
P1	Postulado 1
P2	Postulado 2
P3	Postulado 3
Pr1	Proposição 1
tab.	Tabela
WCCE	World Conference on Computer in Education

Lista de Figuras

Lista de Tabelas

Resumo

O objetivo do presente trabalho é realizar a concepção de um sistema para a aprendizagem de demonstrações da Geometria Euclidiana Plana e a implementação de um protótipo deste sistema, denominado *LEEG - Learning Environment on Euclidean Geometry*, desenvolvido para validar as idéias utilizadas em sua especificação.

Nos últimos anos, tem-se observado uma crescente evolução dos sistemas de ensino e aprendizagem informatizados. A preocupação com o desenvolvimento de ambientes cada vez mais eficientes, tanto do ponto de vista computacional quanto pedagógico, tem repercutido em um salto de qualidade dos software educacionais. Tais sistemas visam promover, auxiliar e motivar a aprendizagem das mais diversas áreas do conhecimento, utilizando técnicas de Inteligência Artificial para se aproximarem ao máximo do comportamento de um tutor humano que se adapte e atenda às necessidades de cada aluno.

A Geometria pode ser vista sob dois aspectos principais: considerada como uma ciência que estuda as representações do plano e do espaço e considerada como uma estrutura lógica, onde a estrutura matemática é representada e tratada no mais alto nível de rigor e formalismo. Entretanto, o ensino da Geometria, nos últimos anos, abandonou quase que totalmente sua abordagem dedutiva. Demonstrações de teoremas geométricos não são mais trabalhadas na maioria das escolas brasileiras, o que repercute em um ensino falho da Matemática, que não valoriza o desenvolvimento de habilidades e competências relacionadas à experimentação, observação e percepção, realização de conjecturas, desenvolvimento de argumentações convincentes, entre outras.

Levando-se em conta este cenário, desenvolveu-se o LEEG, um sistema para a aprendizagem de demonstrações geométricas que tem como objetivo auxiliar um aprendiz humano na construção de demonstrações da Geometria Euclidiana Plana.

O sistema foi modelado sobre uma adaptação do protocolo de aprendizagem MOSCA, desenvolvido para suportar ambientes de ensino informatizados, cuja aprendizagem é baseada na utilização de exemplos e contra-exemplos. Este protocolo propõe um ambiente de aprendizagem composto por cinco agentes, dentre os quais um deles é o aprendiz e os demais assumem papéis distintos e específicos que completam um quadro de ensino-aprendizagem consistente. A base de conhecimento do sistema, que guarda a estrutura lógica-dedutiva de todas as demonstrações que podem ser submetidas ao Aprendiz, foi implementada através do modelo de autômatos finitos com saída. A utilização de autômatos com saída na aplicação de modelagem de demonstrações dedutivas foi extremamente útil por permitir estruturar os diferentes raciocínios que levam da hipótese à tese da proposição de forma lógica, organizada e direta.

As demonstrações oferecidas pelo sistema são as mesmas desenvolvidas por Euclides e referem-se aos Fundamentos da Geometria Plana. São demonstrações que priorizam e valorizam a utilização de objetos geométricos no seu desenvolvimento, fugindo das demonstrações que apelam para a simples manipulação algébrica e que não oferecem uma construção significativa do ponto de vista da Geometria. Porém, mesmo sendo consideradas apenas as demonstrações contidas em *Elements*, todos os diferentes raciocínios para uma mesma demonstração são aceitos pelo sistema, dando liberdade ao aprendiz no processo de construção da demonstração.

Palavras-Chave: Informática na Educação, Demonstração Dedutiva, Ensino de Matemática, Agentes.

TITLE: “A SYSTEM FOR LEARNING OF EUCLIDEAN GEOMETRY’S DEDUCTIVE DEMONSTRATION”

Abstract

This work aims at realizing the definition of a system for Geometry demonstration as well as an implementation of LEEG (Learning Environment on Euclidean Geometry) - a system prototype designed in order to validate the ideas used in its specification.

Teaching and learning systems have constantly evolved over the last few years. The attention given to the development of environments both computationally and pedagogically efficient, has brought about a new standard of educational software. Such systems aim at promoting, assisting with and motivating the learning of a variety of knowledge areas by means of Artificial Intelligence techniques. These techniques are used in order to simulate the behavior of a human tutor which would adapt itself and respond to the needs of each student.

Geometry can be seen from two perspectives: as a science which studies plane and space representations and as a logically structured science, in which its mathematical structure is represented with rigour and formalism.

However, the teaching of Geometry over the last decades has abandoned the deductive approach. Theorem proving has not been taught at most Brazilian schools which renders a deficient teaching of Mathematics. As a consequence, the development of abilities related to experimentation, observation, conjecturing and the development of convincing argumentation has been seriously affected.

Considering this scenario, we have proposed LEEG in order to assist the human learner in the construction of Euclidean Plane Geometry theorem proving.

The system was modeled as an adaptation of the MOSCA learning protocol. MOSCA was developed to support teaching environments, where learning is based on the use of examples and counter-examples. This protocol is based on a teaching environment composed by five agents: the *Aprendiz* and four agents which assume distinct and specific roles in order to build up a consistent teaching and learning framework. We have used the finite automata with output model in order to implement the knowledge base of the system. All logical-deductive structures of the derivations which can be submitted to the *Aprendiz* are stored in this knowledge base.

The use of the finite automata with output in the modeling of deductive demonstrations was extremely useful as it allows the structuring of distinct reasoning patterns from the hypothesis to the thesis proposition in a logical and organized way.

The system provides Euclid's demonstration in the foundations of Plane Geometry. Geometric objects are of great importance in Euclid's demonstrations as opposed to proofs which appeal to algebraic techniques and do not offer a meaningful construction from a geometric viewpoint.

Even though we consider only derivations from the Elements, distinct reasoning for the same demonstration are accepted by the system. Therefore the system provides the learner with a degree of freedom in the demonstration construction process.

Keywords: Informatics in Education, Deductive Demonstration, Mathematics Teaching, Agents.

1 Introdução

A Geometria pode ser vista sob dois aspectos principais [HER 94], especialmente no que diz respeito ao processo de ensino e aprendizagem:

- considerada como uma ciência que estuda as representações do plano e do espaço;
- considerada como uma estrutura lógica, onde a estrutura matemática é representada e tratada no mais alto nível de rigor e formalismo.

Dessa forma, a Geometria tem papel fundamental no entendimento do mundo do real. Por esse motivo, é de extrema importância que se estude sistemas de Inteligência Artificial capazes de representar e raciocinar sobre Geometria.

Na Matemática, a Geometria é a mais antiga área que considera o rigor e o formalismo. O tratamento axiomático da Geometria tem sido investigado por milhares de anos, começando por Euclides, que desenvolveu, a cerca de 2000 atrás, a obra *Elements* ([EUC 78] e [JOY 98]), constituída por 13 volumes, que constrói toda a Geometria através do método axiomático.

"Demonstrações geométricas" é um tema que gera pesquisas tanto na área da Matemática, quanto na área da Ciência da Computação, mais especificamente, em Inteligência Artificial. Essa ligação pode ser explicada pelas semelhanças que tais áreas apresentam, no que diz respeito à estrutura de desenvolvimento da Geometria e à estrutura de aquisição de conhecimento em Inteligência Artificial, que podem ser consideradas semelhantes [NOT 01].

Em Geometria, tem-se os *axiomas*, que determinam proposições evidentes, verdades consideradas óbvias e não demonstráveis e os *teoremas*, que são proposições que, para existirem ou se tornarem evidentes, precisam de demonstração. Na Inteligência Artificial, tem-se uma base de conhecimento, constituída por *predicados básicos* e uma série de predicados que podem, e devem, ser deduzidos a partir dessa base de conhecimento inicial. O que difere uma da outra é que, quando se trata de Matemática, existe um esforço para se produzir um conjunto mínimo de axiomas e, na Inteligência Artificial, é comum e, muitas vezes necessária, a inclusão de axiomas redundantes para facilitar o processo de deduções. Entretanto, isso não afeta a semelhança das estruturas.

Por outro lado, considerando-se o ponto de vista pedagógico, é fundamental que, no processo de ensino e aprendizagem da Geometria, sejam trabalhados e desenvolvidos ambos os enfoques citados acima. Entender a Geometria como uma ciência do espaço é fundamental para o entendimento do mundo real, para a compreensão e ampliação da percepção do espaço e para a representação e visualização de partes do mundo que nos cerca, uma vez que estabelece as relações existentes no plano e no espaço. Por outro lado, compreender sua estrutura rígida e perfeita, ajuda a estruturar o pensamento e o raciocínio dedutivo, desenvolve habilidades de argumentação lógica, de expressão escrita e de precisão da linguagem [PAR 99].

Entretanto, o ensino e aprendizagem da Geometria, nos últimos anos, abandonou quase que totalmente sua abordagem dedutiva. Demonstrações de teoremas geométricos não são mais trabalhadas na maioria das escolas brasileiras, o que repercute num ensino falho da Matemática, que não valoriza o desenvolvimento de habilidades e competências relacionadas à experimentação, observação e percepção, realização de conjecturas, desenvolvimento de argumentações convincentes, entre outras [PAR 99].

Por outro lado, tem-se observado uma crescente evolução dos sistemas de ensino informatizados, que visam promover, auxiliar e motivar a aprendizagem das mais diversas áreas do conhecimento, utilizando técnicas de Inteligência Artificial para se aproximarem ao máximo do comportamento de um tutor humano que se adapte e atenda às necessidades de cada aluno [NOT 00] e [NOT 01c].

Levando-se em conta este cenário, desenvolveu-se um sistema para a aprendizagem de demonstrações geométricas. Em um primeiro momento, pensou-se em formalizar e desenvolver um sistema para a aprendizagem automática, onde seria estudado a evolução do conhecimento de um agente racional artificial na construção de demonstrações de teoremas da Geometria Euclidiana.

Entretanto, mediante a constatação de uma carência de sistemas informatizados de ensino de provas de teoremas geométricos (ver Anexo 3), optou-se em desenvolver o *LEEG - Learning Environment on Euclidean Geometry*, tendo como objetivo, nesta primeira versão, auxiliar um aprendiz humano na construção de demonstrações da Geometria Euclidiana Plana.

As demonstrações oferecidas pelo sistema são as mesmas desenvolvidas por Euclides no primeiro dos treze livros da obra *Elements* (Fundamentos da Geometria Plana) [EUC 78] e [JOY 98]. Optou-se por considerar as demonstrações desenvolvidas por Euclides, por serem demonstrações que priorizam e valorizam a utilização de objetos geométricos no seu desenvolvimento, fugindo das demonstrações que apelam para a simples manipulação algébrica e que não oferecem uma construção significativa do ponto de vista da Geometria.

Porém, mesmo sendo consideradas apenas as demonstrações contidas em *Elements*, todos os diferentes raciocínios para uma mesma demonstração são aceitos pelo sistema, dando liberdade ao aprendiz no processo de construção da demonstração.

O sistema foi modelado sobre o protocolo de aprendizagem MOSCA ([REI 92]), desenvolvido para suportar ambientes de ensino informatizados, cuja aprendizagem é baseada na utilização de exemplos e contra-exemplos. Este protocolo propõe um ambiente de aprendizagem composto por cinco agentes, dentre os quais um deles é o aprendiz e os demais assumem papéis distintos e específicos que completam um quadro de ensino-aprendizagem consistente e suficiente para promover a evolução do conhecimento.

Para uma próxima versão do sistema, está especificada a funcionalidade do agente Aprendiz artificial.

A base de conhecimento do sistema, que guarda a estrutura lógica-dedutiva de todas as demonstrações que podem ser submetidas ao Aprendiz, foi implementada através do modelo de autômatos finitos com saída [MAC 00] e [MAC 01]. A utilização de autômatos com saída na aplicação de modelagem de demonstrações dedutivas foi extremamente útil por permitir estruturar os diferentes raciocínios que levam da hipótese à tese da proposição de forma lógica, organizada e direta. Outra importante vantagem na utilização de autômatos para modelar a base do sistema é a facilidade oferecida na verificação da construção que está sendo desenvolvida pelo Aprendiz, já que cada passo da demonstração é representado por um estado do autômato e cada regra de dedução e representada pela função de transição no autômato. Dessa forma, sabe-se o estado exato em que o Aprendiz se encontra e pode-se associar dicas para cada um desses estados. Ainda se tem a possibilidade guardar o estado onde o Aprendiz parou sua demonstração para, posteriormente, prosseguir a prova do estado em que parou.

Como resultado desta dissertação, os seguintes artigos foram submetidos/publicados a eventos relacionados aos temas do trabalho:

- **Trabalhos publicados em caráter regimental ou regional:**
  1. **Uso do Computador na Educação: Um Histórico Brasileiro**. *Simpósio de Informática do Planalto Médio*, 2., 2000, Passo Fundo. Apresenta uma revisão histórica sobre as políticas educacionais que incentivaram a introdução dos computadores nas escolas, sobre a evolução da utilização dos recursos didáticos utilizados em sala de aula e sobre a acelerada evolução do ensino a distância via Internet, apresentando cursos, projetos, bibliotecas e sistemas de gerenciamento para cursos a distância oferecidos até então. [NOT 00]
  2. **Evolução Cronológica da Informática Educativa na Informática da UFRGS**. *Trabalho Individual I*, 2000. Apresenta uma análise cronológica da evolução dos trabalhos e pesquisas em Informática Educativa desenvolvidos no Instituto de Informática da UFRGS. [NOT 00a]
  3. **Considerações sobre o Raciocínio de um Agente Racional**. *Semana Acadêmica do PPGC*, 2000. É um estudo introdutório sobre a concepção de um agente racional que evolui seu conhecimento em problemas de demonstração em Geometria Euclidiana. [NOT 00b]
- **Trabalhos publicados em Congressos (resumos e trabalhos completos):**
  1. **Hyper-Automaton System Applied to Geometry Demonstration Environment**. *International Conference on Computer Aided Systems Theory and Technology*, 8., 2001, Las Palmas de Gran Canaria. O artigo apresenta um estudo sobre a utilização de Autômatos Finitos com Saída na modelagem da base de conhecimento do sistema de demonstrações em Geometria, verificando a possibilidade de se estruturar uma demonstração axiomática dedutiva através de autômatos finitos. [MAC 01]
  2. **Aprendiz's Learning in Geometry Demonstration**. *World Conference on Computer in Education*, 7., 2001, Copenhagen. O artigo traz as primeiras considerações sobre a concepção do sistema proposto,

apresentado os papéis de cada um dos cinco agentes no ambiente de aprendizagem e um exemplo de demonstração geométrica. [NOT 01]

3. **Historical Overview of Computer usage in Brazilian Teaching**. *World Conference on Computer in Education*, 7., 2001, Copenhagen. É uma versão completa do estudo realizado sobre a evolução da informática no ensino brasileiro, apresentando as políticas governamentais de incentivo à informática na educação, a evolução dos recursos didáticos, das ferramentas de ensino e do ensino a distância via Internet. [NOT 01b]
4. **Knowledge Anticipation on Agents Relationship in the Geometry Demonstration Proof Environment**. *International Conference on Computing Anticipatory Systems*, 5., 2001, Liège. Especifica as interações entre os agentes do protocolo MOSCA aplicados ao LEEG, enfatizando o caráter antecipatório do sistema. Apresenta um exemplo detalhado de interação entre os agentes durante a demonstração de um teorema geométrico, mostrando as possíveis mensagens que são enviadas neste processo. [NOT 01a]
5. **Sistema de Aprendizagem de Demonstrações da Geometria Euclidiana Plana - LEEG**. *Simpósio Brasileiro de Informática na Educação*, 12., 2001, Vitória. Apresenta o sistema LEEG propriamente dito, enfatizando suas características físicas, como interface, documentação, entre outras.

- **Capítulos de livro (to appear 2002):**

1. **Hyper-Automaton System Applied to Geometry Demonstration Environment**. *Lecture Notes in Computer Science*, v. 2178. Versão estendida do artigo. [MAC 01a]
2. **Historical Overview of Computer usage in Brazilian Teaching**. *Pos Conference Book WCCE 2001*. Versão estendida do artigo. [NOT 01c]

- **Trabalhos publicados em Periódicos (to appear 2002):**

1. **Knowledge Anticipation on Agents Relationship in the Geometry Demonstration Proof Environment**. *International Journal of Computing Anticipatory Systems.* Versão estendida do artigo. [NOT 01d]

O volume da dissertação está organizado da seguinte forma:

- **Capítulo 1 -** apresenta o contexto da dissertação, com os objetivos e contribuições do trabalho.
- **Capítulo 2 -** apresenta uma fundamentação teórica, com o objetivo de subsidiar o trabalho do ponto de vista científico. Traz um estudo sobre a Geometria Euclidiana e demonstração de teoremas, Inteligência Artificial, em específico, agentes inteligentes e Autômatos Finitos com Saída.
- **Capítulo 3 -** apresenta o protocolo de aprendizagem MOSCA, especificando os cinco agentes e o esquema de comunicação entre eles. Em seguida, apresenta trabalhos que utilizaram este protocolo na concepção de seus sistemas. Por fim, introduz o sistema LEEG, proposta do presente trabalho, especificando as adaptações que foram feitas sobre o MOSCA para a

aplicação da Geometria Euclidiana Dedutiva. Este capítulo deu origem ao artigo *"Aprendiz's Learning in Geometry Demonstration"*.

- **Capítulo 4 -** apresenta a especificação do sistema LEEG, descreve seu funcionamento, os formalismos que subsidiam sua concepção, a estrutura interna dos dados do sistema e a interação entre os agentes. Apresenta uma classificação interna para as mensagens enviadas pelos agentes durante a aprendizagem das demonstrações. Este capítulo é relacionado aos artigos *"Hyper-Automaton System Applied to Geometry Demonstration Environment"* e *"Knowledge Anticipation on Agents Relationship in the Geometry Demonstration Proof Environment"*.

- **Capítulo 5 -** apresenta considerações sobre a implementação de uma primeira versão do protótipo LEEG, com a descrição da estrutura dos dados, da interface, de características básicas do sistema e as limitações que a versão apresenta. A validação do sistema também é tratada aqui. O artigo *"Sistema de Aprendizagem de Demonstrações da Geometria Euclidiana Plana - LEEG"* está relacionado a este capítulo.

- **Capítulo 6 -** apresenta as últimas considerações sobre o trabalho e especifica os trabalhos futuros relacionados ao tema proposto nesta dissertação e que, por motivo de curto espaço de tempo, ainda não foram trabalhados e desenvolvidos.

Em anexo, ainda encontram-se os artigos que foram publicados em livros ou periódicos em sua versão completa, as 48 proposições de Euclides e a demonstração das cinco primeiras, dentro de uma visão matemática, incluindo desenhos, comentários e suas relações de dependência.

2 Fundamentação Teórica

Este capítulo tem o objetivo de estudar algumas teorias consideradas pré-requisito para o desenvolvimento do presente trabalho. Entre os tópicos abordados no capítulo estão: estudo da Geometria Euclidiana Plana como um sistema axiomático dedutivo; estudo da teoria de Agentes; e estudo de Autômatos Finitos.

2.1 Geometria Euclidiana

Esta seção desenvolve um estudo sobre a Geometria Euclidiana e a classifica como um exemplo clássico de sistema axiomático dedutivo. Apresenta um estudo sobre sistema axiomático e uma introdução sobre os livros desenvolvidos por Euclides de Alexandria em *Elements*.

2.1.1 O Sistema Axiomático

O método axiomático foi inventado pelos matemáticos gregos entre 600 a.C. e 300 a.C. [MEY 74]. Um sistema axiomático, assim como a própria Matemática, é formado por uma coleção de termos primitivos, denominados axiomas, e um conjunto de todos os teoremas que são deduzidos a partir desses termos. Assim, nada, em um sistema axiomático formal, excluindo-se os termos primitivos, pode ser considerado verdadeiro antes de ser provado. Em outras palavras, em um sistema axiomático, parte-se de premissas aceitas como verdadeiras e regras ditas válidas, que conduzem a novas sentenças verdadeiras. As conclusões podem ser alcançadas manipulando-se símbolos de acordo com conjuntos de regras.

Um sistema axiomático (ou matemático) é um sistema abstrato, independente de interpretações sobre seus elementos e suas relações. A Geometria de Euclides é um clássico exemplo de um sistema axiomático, ou seja, sua estrutura só se torna possível através de procedimentos dedutivos que caracterizam um sistema matemático.

Um sistema axiomático dedutivo é composto de termos indefinidos (elementos do sistema), afirmações aceitas como verdadeiras sem necessidade de prova (axiomas, postulados e definições) e afirmações que precisam ser deduzidas e provadas (teoremas) [MES 69]. Todo sistema dedutivo inicia com termos os indefinidos, axiomas, postulados e definições. Com o desenvolvimento do sistema, novos termos são deduzidos a partir das premissas verdadeiras, ou seja, teoremas são provados. Qualquer afirmação logicamente deduzida das afirmações anteriores do sistema (axiomas, postulados, definições, afirmações já provadas) é um teorema.

A primeira definição de qualquer sistema dedutivo deve fazer uso de um ou mais termos indefinidos (elementos) do sistema. O propósito de qualquer definição é tornar possível a comunicação de alguma idéia em poucas palavras.

Para [LAN 91], definição é uma afirmação que requer apenas uma compreensão dos termos a serem usados. Porém, nada se afirma sobre a existência do termo que está sendo definido; isso deve ser provado separadamente. Ou seja, a definição de um termo não implica na sua existência.

Um axioma, segundo o dicionário, é uma proposição evidente, uma proposição não demonstrável cuja aceitação como verdadeira se impõe na formação de uma perfeita seqüência lógica. Para [LAN 91], axioma é uma asserção, a verdade que é tomada como sendo óbvia.

Os postulados, num sistema matemático, constituem as "regras" da matemática [MES 69]. São princípios ou fatos reconhecidos, mas não demonstrados, proposições que são admitidas sem demonstração. Em outras palavras, são assumidos sem provas.

Uma propriedade desejável para um conjunto de postulados é a consistência. Um conjunto de postulados é dito consistente se existe um modelo mínimo para o conjunto.

Uma segunda propriedade de um conjunto de postulados é a independência. Segundo [MES 69], um postulado é independente num conjunto de postulados se existe um modelo do qual todos os outros postulados do conjunto são verdadeiros e o referido postulado é falso.

A independência de postulados em um conjunto de postulados é desejável, mas não é usualmente essencial. Muitas vezes, em cursos mais elementares de matemática, é freqüente a inclusão de postulados adicionais para evitar dificuldades de provas.

Uma vez que tanto os axiomas, quanto os postulados devem ser assumidos como verdadeiros, sem a necessidade de demonstração, pode-se questionar, então, o que difere um do outro. Na verdade, os matemáticos da atualidade já tendem a não diferenciar mais estes dois termos, como os antigos gregos o faziam. Entretanto, Aristóteles apresentou três características que diferem axiomas de postulados [LAN 91]:

- postulados não são evidentes por si só, como os axiomas;
- postulados são aplicáveis somente na ciência específica que está sendo considerada, enquanto que os axiomas são mais gerais;
- postulados afirmam a existência de algo, enquanto os axiomas não.

As proposições são afirmações sobre propriedades de um objeto ou passos para a construção do mesmo. Podem ser teoremas ou problemas de construção, dependendo das afirmações consideradas.

Os teoremas, apesar da necessidade de prova, não provam a existência de nada, isto é, não provam a existência de objetos de um sistema [LAN 91]. Entretanto, os problemas, quando resolvidos, provam a existência dos objetos a que se referem.

Um teorema é uma proposição do tipo $p \rightarrow q$ a qual prova-se ser verdadeira sempre. As proposições p e q são denominadas, respectivamente, hipótese e tese do teorema. Dessa forma, a demonstração de um teorema parte da hipótese e, mediante a aplicação de axiomas e postulados, chega à tese. Isto significa que, dado um teorema, é fundamental, antes de iniciar sua demonstração, identificar claramente a hipótese e a tese.

Ainda encontram-se em um sistema axiomático, corolários e lemas. Um corolário é um resultado que é concluído facilmente de um teorema já provado, ou seja, pode ser considerado uma conseqüência quase direta de um teorema. Em outras palavras, é uma proposição que se deduz imediatamente de outra demonstrada, cuja prova é trivial ou imediata. Um lema, por outro lado, é algo que é assumido na prova de uma proposição, mas que exige confirmação. Os lemas são normalmente produzidos imediatamente antes da conclusão da proposição principal. Em outras palavras, um lema é uma proposição que prepara a demonstração de outra.

2.1.2 Demonstração de Teoremas

Várias são as técnicas de demonstração utilizadas para provar teoremas: prova direta ou dedutiva, prova por contraposição, prova por redução ao absurdo e prova por indução. Entretanto, para este trabalho, interessa apenas a técnica de prova direta ou dedutiva. Neste tipo de prova, pressupõe-se verdadeira a hipótese e, a partir desta, prova-se ser verdadeira a tese.

A construção de uma demonstração dedutiva pode ser associada a, basicamente, duas representações: grafo e textual.

Na primeira representação, tem-se um grafo orientado, onde cada nodo é associado a um enunciado e cada aresta associa um enunciado a outro (regra de dedução). Dessa forma, a representação de uma demonstração sobre a forma de um grafo permite que sua construção mantenha subjacente a estrutura dedutiva dos enunciados.

No grafo, é possível identificar facilmente as relações existentes entre os enunciados, assim como o sentido dessas relações. Outro fator importante é que na demonstração construída num grafo, pode-se visualizar tanto o enunciado do teorema quando sua prova (enxerga-se a inferência e o encadeamento). A fig.2.1 apresenta a representação de uma demonstração na forma de grafo.

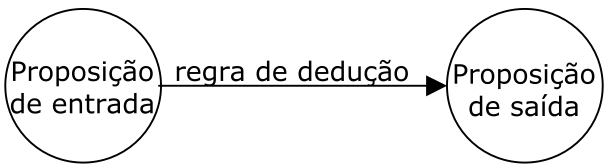

FIGURA 2.1 - Grafo de uma demonstração dedutiva

A representação textual de construção de uma demonstração dedutiva é a forma que possui maior proximidade com o rigor matemático. É desenvolvida seqüencialmente e todos os enunciados devem ser justificados, apresentado coerência e lógica dedutiva. A justificativa para a inclusão de cada enunciado é equivalente à regra de dedução no grafo, que permite a passagem de um enunciado a outro. Ambas as formas de representação - grafo e texto - são equivalentes e válidas para a prova dedutiva de teoremas.

2.1.3 A Estrutura de *Elements* de Euclides

Elements é uma obra desenvolvida por Euclides de Alexandria [EUC 78]. É o mais antigo trabalho matemático grego que foi preservado, onde o método axiomático é altamente desenvolvido em 13 volumes. Seguramente, foi o livro matemático de maior sucesso de todos os tempos, tendo sido utilizado por 2 200 anos.

Nesses 13 livros, Euclides apresenta, de forma extremamente lógica, toda a Geometria Grega Elementar conhecida nos dias de hoje. Isto inclui teoremas e construções da Geometria Plana e Sólida, além da teoria das proporções, da incomensurabilidade e comensurabilidade, teoria dos números e álgebra geométrica.

Como já dito anteriormente, a Geometria é um exemplo clássico de sistema axiomático dedutivo. Dessa forma, *Elements* é organizado e construído a partir de um conjunto de premissas evidentes (axiomas, postulados, definições) e de um conjunto de proposições (teoremas e problemas de construção) que devem ser logicamente provadas a partir dessas premissas.

O Livro I de *Elements* é constituído por 23 definições, 5 postulados, 5 axiomas e 48 proposições, dentre as quais 34 são teoremas e 14 são problemas. Esse volume é relacionado aos fundamentos da Geometria Plana Elementar, incluindo o Teorema de Pitágoras. Nele, Euclides define os termos básicos da Geometria Plana, incluindo ponto, reta, plano, ângulo e figura.

O Livro II é relacionado, sobretudo, à Geometria Algébrica. Entretanto, ainda são dadas soluções para equações quadráticas. É constituído por 2 definições e 13 proposições, onde 12 são teoremas e 1 é problema.

O Livro III é relativo à Geometria do círculo, incluindo o estudo de cordas e setores do círculo. É formado por 11 definições e 37 proposições (31 teoremas e 6 problemas).

O Livro IV discute construções de polígonos regulares de 3, 4, 5, 6 e 15 lados e estuda a construção de figuras inscritas e circunscritas. Possui 7 definições e 16 proposições, todas consideradas problemas.

O Livro V aborda a Teoria das Proporções. Possui 18 definições e 25 proposições. Todas as proposições desse livro são teoremas.

O Livro VI discute as aplicações do Livro V na Geometria Plana, ou seja, estuda figuras semelhantes e proporções. É formado por 11 definições e 37 proposições.

Já os Livros VII, VIII e IX tratam da Teoria dos Números, sendo cada um deles formado por 22 definições e 38 proposições, 27 proposições e 36 proposições, respectivamente.

O Livro X é dedicado aos números irracionais e incomensuráveis. É considerado o mais difícil dos livros dos *Elements* e é constituído por 16 definições e 115 proposições, dentre as quais 91 são teoremas e 24 são problemas.

Por fim, os Livros XI, XII e XIII são centrados na Geometria Sólida (espacial), incluindo o estudo do volume de sólidos familiares. O Livro XI contém o último grupo de definições (28) e 39 proposições (34 teoremas e 5 problemas). O livro XII estuda pirâmides, cones e cilindros e possui 18 proposições, onde 16 são teoremas e duas são problemas. Finalmente, o Livro XIII trata dos cinco sólidos regulares, a saber, tetraedro (composto por quatro triângulos equiláteros), hexaedro (composto por seis quadrados), octaedro (composto por oito triângulos equiláteros), dodecaedro (composto por doze pentágonos regulares) e icosaedro (composto por vinte triângulos equiláteros). Esse livro é formado por 18 proposições, onde 12 são teoremas e 6 são problemas.

2.1.4 Fundamentos da Geometria Euclidiana Plana

Este trabalho é centrado na Geometria Euclidiana Plana. Por esse motivo, vai-se especificar um pouco mais o Livro I de *Elements*, que aborda os Fundamentos da Geometria Plana - teoria dos triângulos, paralelismo e área. Este volume é formado por 23 definições, 5 postulados, 5 axiomas e 48 proposições, que são descritos nas subseções a seguir.

2.1.4.1 Definições

Euclides define termos básicos da geometria plana, como ponto, reta, plano, ângulo e figura. Abaixo, estão as definições que constam no Livro I.

Definição 1: *Um **ponto** é tal que não possui parte.*

Definição 2: *Uma **linha** é tal que não possui largura.*

Definição 3: *As extremidades de uma linha são pontos.*

Definição 4: *Uma linha **reta** é uma linha que possui pontos posicionados uniformemente sobre ela.*

Definição 5: *Uma **superfície** é tal que possui apenas comprimento e largura.*

Definição 6: *As extremidades de uma superfície são linhas.*

Definição 7: *Uma **superfície plana** é uma superfície tal que possui linhas retas posicionadas uniformemente sobre ela.*

Definição 8: *Um **ângulo plano** é a inclinação de duas linhas pertencentes a dois planos que não formam uma linha reta.*

Definição 9: Q*uando as linhas consideradas no ângulo são retas, o ângulo é chamado* ***retilíneo****.*

Definição 10: *Quando uma reta intercepta outra reta e forma ângulos adjacentes iguais, cada um desses ângulos iguais é* ***reto****, e a reta que intercepta a outra é chamada* ***perpendicular*** *à que está interceptando.*

Definição 11: *Um* ***ângulo obtuso*** *é um ângulo maior que um ângulo reto.*

Definição 12: *Um* ***ângulo agudo*** *é um ângulo menor que um ângulo reto.*

Definição 13: *Uma* ***fronteira*** *é a extremidade de qualquer "coisa".*

Definição 14: *Uma* ***figura*** *é tal que está contida por qualquer fronteira ou fronteiras.*

Definição 15: *Um* ***círculo*** *é uma figura plana contida por uma linha tal que todas as retas descendentes de um ponto no interior da figura até tal linha são iguais.*

Definição 16: Este *ponto é chamado* ***centro*** *do círculo.*

Definição 17: *Um* ***diâmetro*** *do círculo é qualquer reta desenhada do centro e terminada em ambas as direções da circunferência do círculo, e tal que a reta bissecta o círculo.*

Definição 18: U*m* ***semicírculo*** *é a figura contida pelo diâmetro e a circunferência cortada por ele. E o centro do semicírculo é o mesmo que o centro do círculo.*

Definição 19: ***Figuras retilíneas*** *são aquelas que são contidas por retas, figuras* ***trilaterais*** *são aquelas contidas por três,* ***quadriláteras*** *são aquelas contidas por quatro e* ***multilaterais*** *aquelas contidas por mais de quatro retas.*

Definição 20: *Das figuras trilaterais, um* ***triângulo equilátero*** *é o que possui seus três lados iguais, um* ***triângulo isósceles*** *é o que possui dois lados iguais e uma* ***triângulo escaleno*** *é o que possui os três lados desiguais.*

Definição 21: *Das figuras trilaterais, um* ***triângulo retângulo*** *é o que possui um ângulo reto, um* ***triângulo obtusângulo*** *é o que possui um ângulo obtuso e um* ***triângulo escaleno*** *é o que possui seus três ângulos agudos.*

Definição 22: *Das figuras quadriláteras, um* ***quadrado*** *é aquele que tem todos os lados iguais e todos os ângulos retos; um* ***retângulo*** *é aquele que tem todos os ângulos retos, mas não lados iguais; um* ***losango*** *é aquele que tem todos os lados iguais, mas não ângulos retos; e um* ***paralelogramo*** *o que tem lados e ângulos opostos iguais, mas nem todos os lados iguais nem todos os ângulos retos. E há outro quadrilátero além desses, chamado* ***trapézio****.*

Definição 23: R*etas* ***paralelas*** *são retas que, estando no mesmo plano, e sendo prolongadas indefinidamente em ambas as direções, não se encontram em nenhuma das direções.*

2.1.4.2 Postulados

Postulado 1: *É possível construir uma reta a partir de qualquer ponto para qualquer ponto.*

Postulado 2: *É possível prolongar uma reta finita continuamente na reta.*

Postulado 3: *É possível descrever um círculo com qualquer centro e raio.*

Postulado 4: *Todos os ângulos retos são iguais entre si.*

Postulado 5: *Se uma reta descendente de duas retas faz ângulos internos no mesmo lado menor que dois ângulos retos, as duas retas, se prolongadas indefinidamente, encontram-se no lado em que os dois ângulos são menores que dois ângulos retos.*

Nos três primeiros postulados, Euclides assume a existência de pontos, linhas e círculos. Isso é necessário porque uma definição, como já dito anteriormente, não implica na existência desses objetos. A existência dos demais objetos geométricos é provada nas proposições posteriores.

O primeiro postulado pode ser interpretado como a afirmação da unicidade da linha reta entre dois pontos dados. Similarmente, o terceiro postulado pode ser interpretado como a afirmação da extensão contínua e infinita do espaço, isto é, o raio pode ser indefinidamente pequeno, o que implica que não existe distância mínima entre dois pontos no espaço (por isso o espaço é contínuo), e, por outro lado, o raio pode ser indefinidamente grande e então não existe distância máxima entre dois pontos no espaço.

O quarto e quinto postulados foram, durante muito tempo, pensados e tratados como teoremas que poderiam ser provados.

2.1.4.3 Axiomas

Axioma 1: *Objetos que são iguais a um mesmo objeto, são iguais um ao outro.*

Axioma 2: *Se igualdades são adicionadas a igualdades, então o total é igual.*

Axioma 3: *Se igualdades são subtraídas de igualdades, então o restante é igual.*

Axioma 4: *Objetos que coincidem um com outro, são iguais um ao outro.*

Axioma 5: *O todo é maior que a parte.*

Dentre os cinco axiomas, o quarto é o que exige maior esclarecimento: esse axioma afirma que o método da superposição é aceitável para a prova de igualdade de duas figuras.

2.1.3.4 Proposições

Neste livro, Euclides apresenta a geometria das retas e ângulos no plano, incluindo resultados de triângulos, retas que se interceptam, retas paralelas e paralelogramos.

Proposição 1: C*onstruir um triângulo equilátero sobre um segmento de reta dado.*

Proposição 2: *Construir um segmento de reta igual a um segmento de reta dado com uma das extremidades em ponto dado.*

Apresentou-se aqui apenas as duas primeiras proposições, por serem as proposições que foram de fato implementadas na primeira versão do protótipo LEEG. As demais proposições podem ser encontradas no Anexo 1 deste volume.

As três primeiras proposições referem-se às três operações fundamentais da Geometria Euclidiana. A maioria das demais proposições é dependente dessas construções.

2.2 Teoria de Agentes

Esta seção apresenta um estudo sobre a teria de Agentes Inteligentes, com alguns conceitos e definições encontrados na bibliografia e a classificação dos agentes em função de suas propriedades e funcionalidades.

2.2.1 Conceitos e Definições

A discussão sobre o que vem a ser uma entidade racional ou inteligente apresenta diferentes posicionamentos dos mais diversos pesquisadores da comunidade científica. A tentativa de se encontrar uma única definição para o que se chama de inteligência ainda fracassa. Entretanto, existe um consenso: entidades que apresentam determinado comportamento, considerado inteligente, são os chamados *agentes*.

Muitas são as definições para agentes. [RUS 95] apresenta algumas definições, de acordo com o aumento de sofisticação ou de inteligência:

- no que diz respeito ao poder de sensibilidade e percepção, um agente é algo que pode ser visto e percebido por seu ambiente através de sensores e agir sobre ele através de reatores;

- com o acréscimo da racionalidade, para cada situação de percepção possível, um agente racional ideal deve maximizar sua performance, com base na evidência produzida pela percepção e no conhecimento construído que já possui;
- um agente racional pode, num próximo nível de comportamento inteligente, agir autonomamente. Ou seja, um agente autônomo deve executar suas ações e escolhas em função de sua própria experiência.

[FRA 96] apresenta a seguinte definição: um agente autônomo é um sistema que compõe parte de um ambiente, é percebido por ele e age sobre ele, o tempo todo, perseguindo seus próprios objetivos.

Em [KNA 98], encontram-se ainda outras definições para agentes inteligentes: agentes inteligentes são software ou entidades que trazem consigo um conjunto de operações de interesse do usuário ou de programa, com certo grau de independência ou autonomia e empregam conhecimento ou representação para alcançar as metas e desejos. Agentes inteligentes podem, então, ser descritos em termos de um espaço definido por essas duas dimensões: agência e inteligência.

Agência é o grau de autonomia e autoridade assumida por um agente e pode ser medida, ao menos qualitativamente, pela natureza da interação do agente com outras entidades do sistema. No mínimo, um agente deve rodar assincronamente. O grau de agência é aumentado se um agente representa um usuário de determinada forma. Esta é uma das principais características dos agentes. Um agente mais avançado pode interagir com outras entidades, como dados, aplicações ou serviços. Agentes ainda mais avançados colaboram e negociam com outros agentes.

Alguns atributos são discutidos em [WOO ??], dentro do contexto de agência, tais como *mobilidade*, que é a habilidade de um agente de mover-se em uma rede de computadores; *veracidade*, que é a suposição de que um agente não comunicará falsas informações; *benevolência*, que é a suposição de que agentes não terão metas conflitantes; e *racionalidade*, que é a suposição de que um agente agirá em função de alcançar seus objetivos.

Inteligência é o grau de raciocínio e comportamento aprendido pelo agente. A habilidade do agente em aceitar metas do usuário e executar as tarefas delegadas para isso determina o grau de inteligência do agente. No mínimo, ele pode estabelecer preferências, na forma de regras, com inferências ou outro mecanismo para agir sobre essas preferências. Níveis mais avançados de inteligência incluem modelo de usuário ou alguma outra forma de compreender e raciocinar sobre os desejos do usuário. Uma escala ainda mais evoluída de inteligência é apresentada por sistemas que aprendem e adaptam-se ao ambiente que estão inseridos, ambos em termos dos objetivos do usuário e em termos dos recursos disponíveis para o agente. Tais sistemas descobrem novas relações, conexões ou conceitos independentemente do usuário humano, e exploram-nas para antecipar e satisfazer suas necessidades.

Uma definição mais geral, retirada de [KNA 98], pode ser expressa como segue:

- *Autonomia* - agentes operam sem intervenção direta de humanos ou de outros agentes e possuem controle sobre seu estado interno;

- *Habilidade Social* - agentes interagem com outros agentes, artificiais ou humanos, através de alguma linguagem de comunicação dos agentes;
- *Reatividade* - agentes percebem seu ambiente e respondem em tempo a mudanças que ocorrem nele;
- *Proatividade* - agentes não simplesmente agem em resposta a seu ambiente, mas são capazes de exibir comportamento objetivo através de iniciativas.

Ainda é possível definir os agentes inteligentes pelo domínio em que produzem seus serviços:

- busca de informação;
- filtro de dados;
- monitoramento de condições e alertas quando um conjunto de determinadas condições é detectado;
- performance de ações em favor do usuário;
- produção de *help* automático, tutor on-line;
- gerência de redes;
- existência de atores virtuais e atuação em filmes;
- utilização de sistemas otimizados através de técnicas de metas diretas.

Uma vez que as tarefas apontadas acima são delegadas com sucesso, os agentes podem ser poderosas ferramentas para administração e gerência de sistemas computacionais complexos.

Este novo enfoque para os sistemas inteligentes abre um novo espectro de inteligência. Muitos dos sistemas inteligentes até então conhecidos e utilizados eram limitados, pois usavam um conjunto limitado de regras, muitas vezes inconsistentes, e possuíam pouca ou nenhuma autonomia e flexibilidade. Num outro extremo, os mais avançados tipos de agentes que agora emergem, possuem comportamento chamado inteligente. Esse comportamento apresentado pelos agentes é associado a entidades inteligentes, conhecidas como "humanos". Estes agentes são capazes de apresentar tal comportamento porque incorporam várias técnicas de Inteligência Artificial.

Com tecnologias computacionais avançadas, o advento de sistemas cada vez mais complexos, distribuídos e heterogêneos tem implicado na dificuldade de desenvolvimento de software utilizando métodos de programação tradicionais. Com isso, vem-se incorporando a utilização de agentes em aplicações de sistemas distribuídos.

No que diz respeito à compreensão de símbolos pelas máquinas que apresentam comportamento inteligente, tenta-se representar corretamente fatos sobre o mundo através de uma linguagem sintática (configuração de símbolos) e semântica (símbolos que representam os fatos; intenção de uma afirmação). Adicionalmente, tenta-se raciocinar sobre esses fatos, usando lógica ou outros mecanismos para gerar inferências, isto é, deduzir novos fatos ou verificar se os fatos são verdadeiros ou falsos.

Muitas técnicas de Inteligência Artificial podem ser utilizadas para auxiliar um agente na interpretação da "fala" de outros agentes, humanos ou artificiais. Essas técnicas incluem:

- *Análise sintática* - uso da estrutura de uma sentença junto com uma palavra na seqüência e referir preferências para resolver problemas de ambigüidade;
- *Análise léxica* - resolve problemas de ambigüidade pela busca do sentido da palavra é mais provável em determinados contextos;
- *Análise semântica* - usa contexto e o conhecimento já armazenado na base de conhecimento;
- *Aplicação apropriada da base de conhecimento* - linguagens coloquiais, figuras de linguagem, metáforas, comparações, homônimos, sinônimos, etc.

Uma das aplicações de sistemas agentes, é que muitos agentes autônomos construídos com técnicas de Inteligência Artificial podem colaborar para resolver problemas complexos. Em sistemas colaborativos, cada agente contribui com sua solução particular para o problema, com o objetivo de solucionar o problema mais complexo.

Muitas são as áreas em que a Inteligência Artificial clássica vem tratando na construção e entendimento de "máquinas que pensam". São elas:

- *Resolução de problemas* - busca num espaço grande de problemas e redução de problemas;
- *Raciocínio lógico* - dedução, prova de teoremas, análise de base de dados e manutenção;
- *Compreensão de linguagem* - tradução, leitura de texto, compreensão de palavras e expectativas baseadas no contexto;
- *Programação automática* - sistemas que podem escrever programas para arquivar alguns resultados baseados em descrições da proposta;
- *Aprendizagem* - progressos relativamente recentes usando redes neurais e algoritmos genéticos;
- *Base de conhecimento e sistemas expertos* - uma das maiores aplicações das tecnologias de IA. Engenharia do conhecimento converte uma perícia humana dentro de um domínio de conhecimento para formas compreensíveis por outros;
- *Linguagens, sistemas operacionais e ferramentas* - abrangendo linguagens gerais de IA, como LISP e Prolog, para linguagens específicas para agentes como Telescript. Linguagens orientadas a objetos e ambientes distribuídos tem-se tornado fatores importantes no desenvolvimento de agentes.

2.2.2 Classificação de Agentes

As diversas definições encontradas para agentes envolvem uma série de propriedades que caracterizam um agente. Tais propriedades acabam por diferenciar os diferentes agentes e classificá-los em função de suas aplicações. A tab.2.1 mostra algumas propriedades consideradas anteriormente.

TABELA 2.1- Classificação dos Agentes

Propriedade	Outras denominações	Significado/Intenção
Reativo	Sensitivo e ativo	Responde em tempo hábil a mudanças do ambiente
Autônomo		Exercício de controle sobre suas próprias ações
Orientado por meta	Pró-ativo Propositado	Não age somente em resposta ao ambiente
Contínuo		Está continuamente rodando, processando
Comunicativo	Habilidade social	Comunica-se com outros agentes, inclusive humanos
Aprendiz	Adaptativo	Muda seu comportamento baseado em experiências anteriores
Móvel		Capaz de transportar-se sozinho de uma máquina para outra
Flexível		Ações não são escritas
Caráter		Possível personalidade e estado emocional

2.3 Autômato Finito

Autômato Finito pode ser considerado uma máquina de estados, composta basicamente de três partes: *fita*, que é o dispositivo de entrada que contém a informação a ser processada; *unidade de controle*, que reflete o estado corrente da máquina; e a *função de transição*, que comanda as leituras e define o estado da máquina [MEN 00].

Um autômato é representado por uma 5-upla:

$$M = (\Sigma, Q, \delta, q_0, F)$$

onde:

Σ alfabeto de símbolos de entrada;
Q conjunto de estados possíveis do autômato, o qual é finito;
δ função programa ou função de transição, a qual é uma função parcial;

$$\delta: Q \times \Sigma \to Q$$

q_0 estado inicial do autômato, tal que q_0 é elemento de Q;
F conjunto de estados finais, tal que F está contido em Q;

A função de transição de um Autômato Finito pode ser representada por um grafo finito direto, como mostra a fig.2.2, e pode ser manipulado sob vários aspectos.

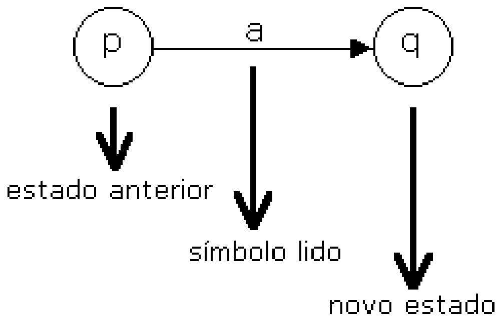

FIGURA 2.2 - Representação da função de transição como um grafo

O processamento de um Autômato Finito consiste na sucessiva aplicação da função de transição para cada símbolo da palavra de entrada até ocorrer uma condição de parada [MEN 00].

2.3.1 Autômato Finito com Saída

Para a aplicação deste trabalho, entretanto, é preciso trabalhar com uma modificação do conceito de Autômato Finito, que permite a geração de uma palavra de saída associada aos estados ou às transições (Máquinas de Moore e Mealy, respectivamente) [HOP 79, MEZ 00]. Ambas as máquinas são definidas sobre um alfabeto especial, denominado Alfabeto de Saída, e o resultado do processamento é o seu estado final (condição de aceita/rejeita) e a informação contida na fita de saída.

2.3.1.1 Máquina de Moore

A Máquina de Moore é um Autômato Finito com Saída, que gera uma palavra de saída para cada estado da máquina. É representada por uma 7-upla:

$$M = (\Sigma, Q, \delta, q_0, F, \Delta, \delta_S)$$

onde:

Σ alfabeto de símbolos de entrada
Q conjunto de estados possíveis do autômato, o qual é finito;
δ: $Q \times \Sigma \rightarrow Q$ função programa ou função de transição, a qual é uma função parcial que, dependendo do estado corrente e do símbolo lido, determina o novo estado do autômato;
q_0 estado inicial tal que q_0 é elemento de Q;
F conjunto de estados finais tal que F está contido em Q;
Δ alfabeto de símbolos de saída;
δ_S: $Q \rightarrow \Delta^*$ função de saída, a qual é uma função total que determina a geração de uma palavra de saída para cada estado.

A função de transição pode ser representada como um grafo direto, como no Autômato Finito, adicionando, na etiqueta de cada estado, a saída associada (fig.2.3). Possui uma segunda função (função de saída), que gera uma palavra de saída para cada estado da máquina.

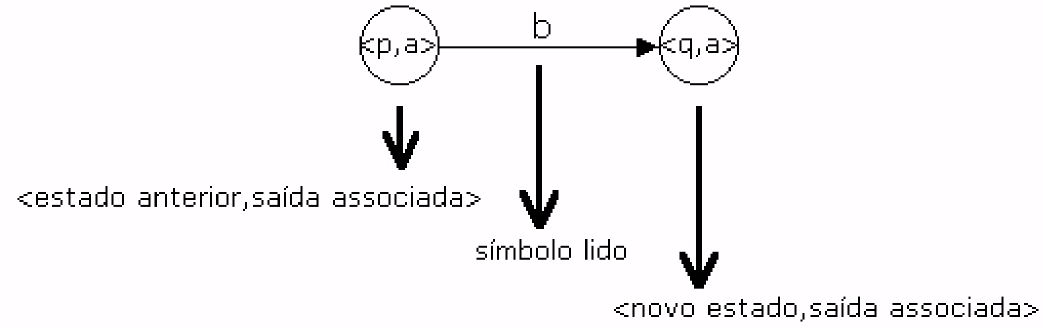

FIGURA 2.3 - Grafo da função de transição da Máquina de Moore

2.3.1.2 Máquina de Mealy

A Máquina de Mealy é um autômato finito modificado de forma a gerar uma palavra de saída para cada transição. É representada por uma 6-upla:

$$M = (\Sigma, Q, \delta, q_0, F, \Delta)$$

onde:

Σ alfabeto de símbolos de entrada;
Q conjunto de estados possíveis do autômato, o qual é finito;
δ: Q x $\Sigma \rightarrow$ Q x Δ^* função programa ou função de transição, a qual é uma função parcial que, dependendo do estado corrente e do símbolo lido, determina o novo estado do autômato e a saída gerada;
q_0 estado inicial tal que q_0 é elemento de Q;
F conjunto de estados finais tal que F está contido em Q;
Δ alfabeto de símbolos de saída.

A função de transição pode ser representada por como um grafo finito como no Autômato Finito, adicionando apenas, na etiqueta de cada transição, a saída associada. A fig.2.4 ilustra o grafo da função de transição de uma Máquina de Mealy.

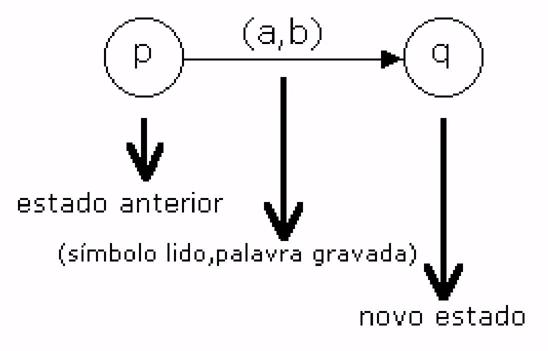

FIGURA 2.4 - Grafo da função de transição da Máquina de Mealy

A especificação sobre as correspondências entre as n-uplas do autômato e a base de conhecimento do sistema proposto no trabalho será tratada mais adiante, no capítulo 4.

2.4 Considerações Finais

O presente capítulo teve o objetivo de realizar um estudo sobre as teorias necessárias para o desenvolvimento e fundamentação deste trabalho. Entre as teorias aqui estudadas, estão:

- *Geometria Euclidiana* - domínio de aplicação da proposta, juntamente com a área de demonstração de teoremas;
- *Teoria de Agentes* - necessária para concepção do sistema;
- *Autômato Finito* - formalismo utilizado para modelagem e implementação da base de conhecimento do sistema.

O capítulo que segue apresenta um estudo sobre o protocolo de aprendizagem MOSCA e os principais sistemas que o utilizaram em suas concepções. Logo a seguir, é apresentado o sistema LEEG (*Learning Environment on Euclidean Geometry*), proposta deste trabalho.

3 Protocolo de Aprendizagem MOSCA e Sistemas Derivados

Este capítulo está diretamente relacionado ao artigo *"Aprendiz's Learning in Geometry Demonstration"* ([NOT 01]), que faz considerações iniciais sobre a concepção do sistema proposto nesse trabalho.

Num primeiro momento, pretende-se definir o quadro teórico que sustenta o sistema proposto no trabalho. As teorias apresentadas aqui foram selecionadas por enquadrarem-se na proposta de construção de um sistema de prova que faz evoluir o conhecimento através de interações entre agentes, humanos ou artificiais, e permite o controle da construção e evolução da aprendizagem.

3.1 Protocolo MOSCA

Algumas teorias formais de aprendizagem definem ou estabelecem que um ambiente mínimo de aprendizagem de máquina deve ser composto por um aprendiz, que se comunica com um oráculo para promover a aprendizagem.

O funcionamento básico desse ambiente mínimo de aprendizagem é o seguinte: o Oráculo envia ao Aprendiz problemas resolvidos que são memorizados por ele e constituem sua amostra de exemplos. Tendo esses exemplos, o Aprendiz procura, dentro de um espaço de hipóteses, aquela que melhor se amolda aos exemplos enviados pelo Oráculo. Isto se dá pela introdução de um critério que define qual a melhor hipótese em relação aos exemplos oferecidos. Esse critério é chamado aprendizagem. Dessa forma, o espaço de hipóteses do Aprendiz é o conjunto de todo o conhecimento que ele pode aprender.

Na prática, esse ambiente mínimo de aprendizagem automática concretiza-se da seguinte forma: a máquina assume o papel de Aprendiz e o *expert*, o papel de Oráculo. Todas as escolhas cruciais para a aprendizagem são de responsabilidade do Oráculo: o quadro teórico subjacente à aprendizagem de um problema, ou seja, tipo de Aprendiz, forma das hipóteses, dos exemplos e dos problemas a resolver, e a seleção dos exemplos. O Oráculo ainda dispõem de um primeiro meio de pressão para impor ao Aprendiz determinado conhecimento.

Depois de ter recebido os primeiros exemplos e escolhido uma hipótese, o Aprendiz está apto a resolver o problema proposto. Não se pode, no entanto, avaliar a hipótese escolhida pelo Aprendiz com base na solução apresentada por ele. Para permitir que o *expert* explorasse a hipótese escolhida pelo Aprendiz, [REI 92] introduziu ao Aprendiz um mecanismo de argumentação, ou seja, a pedido do *expert*, o Aprendiz argumenta seus resultados. Uma crítica à argumentação do Aprendiz o fará revisar a hipótese que escolheu.

Este ambiente de aprendizagem proposto por [REI 92], denominado MOSCA (Mestre + Oráculo + Sonda + Cliente + Aprendiz), está ilustrado na fig.3.1. Ele é composto por cinco entidades distintas (considerados agentes humanos ou artificiais) que apresentam comportamentos específicos de acordo com os objetivos propostos e interagem no processo de aprendizagem. É um protocolo de aprendizagem baseado na aprendizagem através de exemplos. Uma breve descrição dessas cinco entidades é apresentada a seguir:

- ***Aprendiz*** - dispõem de uma hipótese, escolhida dentre de um espaço de hipóteses, que é adequada ao conjunto de exemplos já memorizados;
- ***Oráculo*** - produz problemas resolvidos cujas soluções são irrefutáveis;
- ***Sonda*** - da mesma forma que o Oráculo, produz problemas resolvidos, porém com solução refutáveis, ou seja, que podem e devem ser contestadas pelo Aprendiz. Seu principal objetivo é obrigar o Aprendiz a argumentar;
- ***Cliente*** - submete ao Aprendiz um problema a ser resolvido e aguarda sua solução;
- ***Mestre*** - analisa a argumentação do Aprendiz e envia uma crítica para cada argumento produzido por ele.

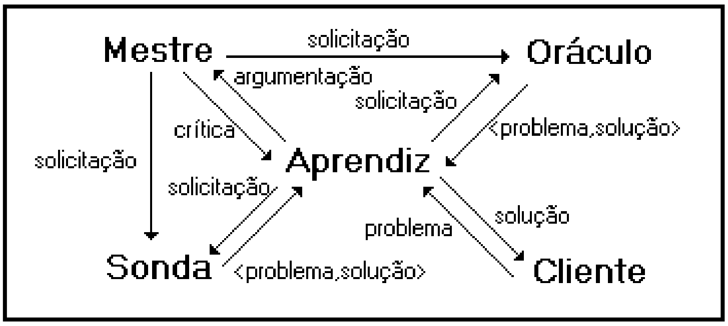

FIGURA 3.1 - Protocolo de aprendizagem MOSCA

O processo inicia com a submissão de um problema a ser resolvido pelo Cliente ao Aprendiz. O Aprendiz, então, solicita do Oráculo um conjunto de exemplos, que serão memorizados por ele. Toda a transformação nesse conjunto de exemplos formado pelo Aprendiz deve conduzir a uma revisão da hipótese escolhida. Esta hipótese, como já dito anteriormente, é retirada de um espaço de hipótese e deve satisfazer a um critério de aprendizagem. Na prática, a hipótese escolhida deve ser capaz de encontrar as soluções para os problemas apresentados no conjunto de exemplos.

O Aprendiz pode ainda solicitar do Oráculo a aprovação de um problema resolvido. Essa solicitação se reduz a um simples sinal.

O Aprendiz recebe também problemas resolvidos do Sonda. Entretanto, tais soluções podem ser incorretas. O Aprendiz compara sua solução (obtida a partir da hipótese escolhida) com a solução apresentada pela sonda e argumenta sua resolução ao Mestre. Tal argumentação pode ser de duas formas distintas: uma explicação, caso as duas soluções coincidam, ou uma objeção, caso as duas soluções sejam diferentes. Uma vez que a função de argumentação é dependente da hipótese escolhida pelo Aprendiz, toda alteração na hipótese deve conduzir a uma revisão da argumentação.

Para cada argumentação produzida pelo Aprendiz, uma crítica é enviada pelo Mestre. As argumentações criticadas são memorizadas pelo Aprendiz. Toda crítica negativa leva o Aprendiz a apresentar uma nova argumentação. Caso não seja possível produzir uma nova argumentação, o Aprendiz deve rever sua hipótese, solicitando novos exemplos ao Oráculo.

Essas argumentações são retiradas de um espaço de argumentos.

O Mestre controla a produção de problemas resolvidos do Sonda através de solicitações, que podem ser na forma de sinais ou de problemas específicos que quer que submeta.

O Mestre ainda pode forçar o Aprendiz a argumentar, quando esse não está sendo capaz de produzir argumentos, através de solicitações ao Oráculo para o envio de mais exemplos ao Aprendiz. Da mesma forma que anteriormente, essas solicitações podem ser na forma de sinais ou de problemas específicos.

A característica central desse modelo de aprendizagem é que as interações entre os agentes são concebidas sobre a forma de objetos do conhecimento.

Pode-se perceber que sua concepção está fundamentada na teoria de aprendizagem construtivista, uma vez que a aprendizagem é construída pelo conflito cognitivo.

O protocolo de aprendizagem MOSCA foi utilizado em alguns sistemas de ensino-aprendizagem já desenvolvidos, com diferentes enfoques e abordagens. Vai-se, a seguir, apresentar algumas dessas abordagens de utilização do modelo consideradas interessantes e relevantes para o trabalho.

3.2 SAID-Cabri

O sistema de aprendizagem automática *SAID-Cabri*, desenvolvido em [FER 92] utilizou esse protocolo na concepção de um agente Aprendiz. Dentre as cinco entidades propostas por [REI 92], apenas uma foi automatizada. Assim, nessa abordagem, o aprendiz foi considerado como sendo um agente artificial, e as demais entidades (Oráculo, Sonda, Cliente e Mestre) foram representadas pelo mesmo agente humano.

O objetivo central da tese foi desenvolver um agente racional capaz de aprender conceitos de geometria, através do diálogo entre agentes e construindo um conhecimento argumentável, com base no conhecimento já adquirido anteriormente.

O trabalho foi desenvolvido no ano de 1992 por Edilson Ferneda em sua tese de doutorado, na Universidade de Montpellier II, na França. A fig.3.2 apresenta a abordagem do MOSCA utilizada por [FER 92].

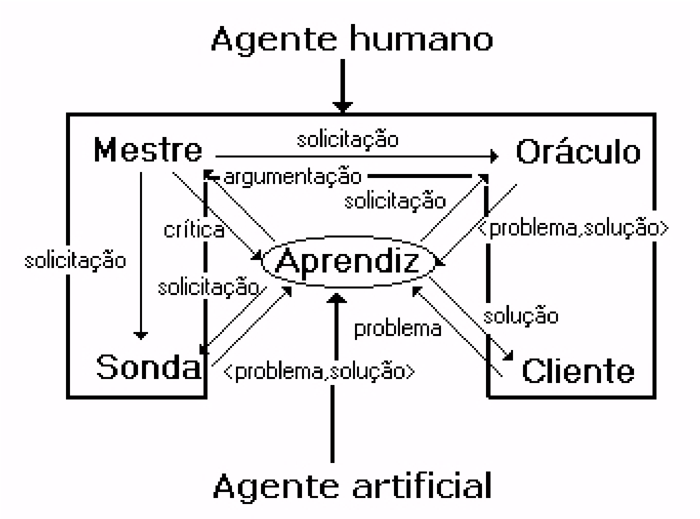

FIGURA 3.2 - Abordagem do Protocolo MOSCA utilizado em [FER 92]

3.3 Cabri-Euclide

O micromundo de prova *Cabri-Euclide* [LUE 97], desenvolvido por Vanda Luengo em sua tese de doutorado, na Universidade Joseph Fourier - Grenoble I, França, no ano de 1997, também utilizou o modelo de aprendizagem de Reitz na concepção do sistema. Nesta abordagem, o Aprendiz não é mais considerado como um agente artificial e automático, mas sim como um agente humano, mais especificamente, um aluno. O Mestre, o Sonda e o Oráculo constituem o meio, ou seja, a parte automatizada do sistema.

O objetivo do trabalho foi desenvolver um ambiente de aprendizagem humana que permitisse a aprendizagem da prova em geometria integrando a refutação no processo. Para que isso acontecesse, os agentes artificiais foram projetados para produzir contradições, dificuldades e desequilíbrios através de críticas e retroações emitidas a cada enunciado inconsistente do Aprendiz. As interações entre o Aprendiz e o meio são da forma de ações, formulações e validações. A fig.3.3 ilustra essa abordagem do modelo proposto por Reitz.

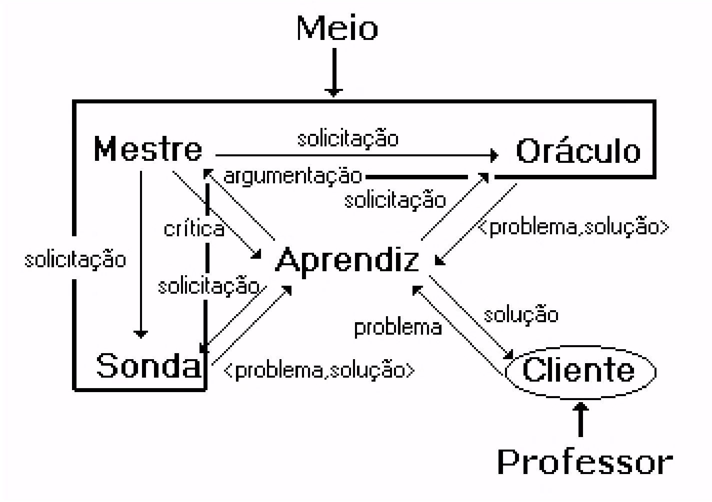

FIGURA 3.3 - Abordagem do Protocolo MOSCA utilizado em [LUE 97]

3.4 MATHEMA

O ambiente de aprendizagem mediado por computador MATHEMA, desenvolvido em [COS 95], na Universidade Federal da Paraíba, propõem novamente uma pequena modificação na estrutura inicial do protocolo MOSCA: as entidades Mestre, Oráculo e Sonda são consideradas como sendo um tutor artificial enquanto que as entidades Cliente e Aprendiz são consideradas professor e aluno (ambos humanos), respectivamente.

Conceitualmente, o agente tutor artificial é composto de dois níveis: um nível externo e um nível interno. No nível externo, é localizado o Mestre, que interage diretamente com o Aprendiz. No nível interno, tem-se o Oráculo e o Sonda, que são consideradas entidades passivas que fornecem conhecimento para o Mestre. A fig.3.4 mostra um modelo mínimo do sistema MATHEMA.

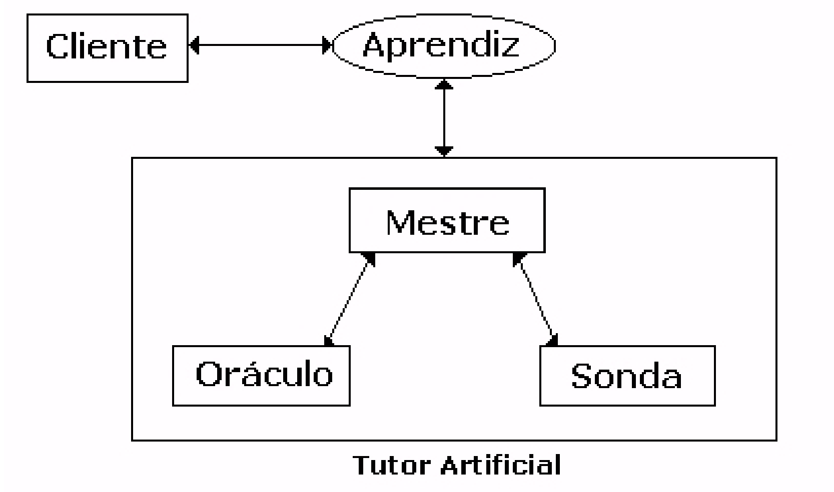

FIGURA 3.4 - Abordagem do Protocolo MOSCA utilizado em [COS 95]

3.5 Sistema LEEG

Baseando-se nos projetos que já foram desenvolvidos com a utilização do protocolo de aprendizagem MOSCA, propõe-se uma nova abordagem dessa teoria, que será aplicada no sistema LEEG (*Learning Environment on Euclidean Geometry*).

O LEEG é um ambiente para a aprendizagem de Geometria Euclidiana Dedutiva que tem como objetivo auxiliar o processo de construção de demonstrações de teoremas (proposições) da Geometria Euclidiana Plana.

O sistema propõe a aprendizagem de construção de demonstrações geométricas através do auxílio de exemplos e contra-exemplos, que caracterizam intervenções do sistema referentes à inconsistência de afirmações ou à utilização incorreta dos termos que formam a Geometria Euclidiana.

Por ser a Geometria Euclidiana um exemplo clássico de um sistema axiomático, sua estrutura de formação é deduzida de um conjunto de premissas básicas que originam as demais proposições que as procedem [MEY 74].

Como já mencionado anteriormente, no capítulo 2 (seção 2.1.4), os termos envolvidos em um sistema axiomático são:

- ***Definições*** **-** afirmações que apenas requerem uma compreensão dos termos que são empregados na demonstração;
- ***Postulados*** **-** princípios ou fatos reconhecidos, mas não demonstrados, isto é, admitidos sem demonstração;
- ***Axiomas*** **-** proposições evidentes e não demonstráveis;
- ***Proposições*** **-** afirmações sobre propriedades de objetos (*teoremas*) ou passos para sua construção (*problemas*), que devem ser demonstradas para serem aceitos como enunciados verdadeiros.

Em outras palavras, as definições, postulados e axiomas formam o conjunto de termos evidentes de um sistema axiomático, que são admitidos como verdadeiros sem necessidade de prova. As proposições devem ser provadas, a partir dos termos evidentes e das regras que estabelecem o sistema. Sempre que uma proposição é demonstrada, pode-se admiti-la como verdadeira e utilizá-la na demonstração de outras proposições posteriores, ou seja, ela passa a compor o conjunto de enunciados verdadeiros.

Em uma proposição, deve-se identificar a hipótese (ou as hipóteses) e a tese. No caso da prova dedutiva, a hipótese da proposição é tomada como verdadeira e deve ser utilizada para a construção da prova. A tese, por sua vez, é o enunciado que deve ser provado, através de uma seqüência lógica e rígida de enunciados, formada pelas afirmações evidentes ou já demonstradas.

Uma demonstração será, então, um conjunto de enunciados matemáticos, que devem ser justificados pelo uso de definições, postulados e axiomas, além das proposições já demonstradas anteriormente. Estes enunciados devem estar rigorosamente estruturados e ordenados de forma lógica e hierárquica. A rigidez na estrutura da demonstração deve ser obedecida para que se reproduza uma prova dentro de um sistema axiomático dedutivo.

Assim, o sistema LEEG objetiva auxiliar um agente humano (Aprendiz) na construção de provas da Geometria Euclidiana Plana, tendo como base o rigor de um sistema axiomático. A construção das demonstrações desenvolvida pelo Aprendiz deve obedecer a uma seqüência lógica de formação. Cada passo desenvolvido por ele é acompanhado pelo agente Mestre, que identifica inconsistências e incorreções, e ativa o envio de mensagens (exemplos e contra-exemplos) que devem intervir na sua construção.

3.5.1 Agentes

Uma descrição dos papéis assumidos por cada um dos agentes no sistema LEEG é apresentada a seguir:

- ***Mestre*** - é o agente que constantemente acessa a construção da demonstração que está sendo desenvolvida pelo Aprendiz, verificando a sua estrutura. A verificação é feita através da comparação entre a estrutura desenvolvida pelo Aprendiz e a estrutura contida na base de conhecimento do ambiente. Essa comparação é feita passo a passo, a cada nova afirmação e ligação incluída na demonstração;
- ***Oráculo -*** é o agente que, em conformidade com as sinalizações do Mestre, interage com o Aprendiz através de exemplos irrefutáveis, objetivando dar subsídios para o Aprendiz formular enunciados e estabelecer as regras de dedução durante o processo de elaboração da demonstração. Os exemplos são armazenados em uma base de exemplos. A seleção dos exemplos é determinada pela sinalização do Mestre, que identifica o estágio da demonstração e o erro cometido pelo Aprendiz;
- ***Sonda -*** o agente Sonda tem a função de enviar ao Aprendiz, de acordo com as solicitações do Mestre, exemplos que podem ser refutados. Estes exemplos são enviados quando o Mestre identifica erros de regras de dedução ou enunciados da demonstração produzidos pelo Aprendiz. Com os contra-exemplos, o Aprendiz deve refazer o passo da construção onde foi identificado o erro para poder prosseguir com a demonstração;
- ***Cliente -*** é o agente que inicia o processo através da submissão de uma das possíveis proposições ao agente Aprendiz que deve ser demonstrada. O acesso à base de proposições pode ser efetuado de duas maneiras: ordenada ou randômica. Essas duas formas estabelecem duas abordagens na construção das demonstrações dedutivas. A primeira consiste em um *raciocínio indutivo*, ou seja, não são propostas ao Aprendiz proposições que exijam resultados de proposições ainda não demonstradas. A segunda abordagem caracteriza-se por um *raciocínio recursivo*, ou seja, podem ser

propostas proposições que dependem do resultado de outras proposições que ainda não foram provadas pelo Aprendiz;

- ***Aprendiz*** - o agente Aprendiz recebe do Cliente uma proposição para ser demonstrada. Numa proposição, deve-se identificar a hipótese e a tese. No caso de provas dedutivas, a demonstração deve partir da hipótese, que é admitida como verdadeira, e provar a tese, através de uma seqüência lógica de enunciados que devem ser rigorosamente justificados. O papel do Aprendiz é exatamente este: provar ser verdadeira a tese da proposição que foi submetida a ele, através de uma seqüência de afirmações justificadas. Cada passo do seu raciocínio é acompanhado pelo agente Mestre, que interrompe o processo sempre que identifica um erro de afirmação ou ligação, e solicita dos agentes Oráculo e Sonda a emissão de exemplos ou contra-exemplos, dependendo da situação identificada.

A fig.3.5 mostra o protocolo MOSCA adaptado para o sistema LEEG.

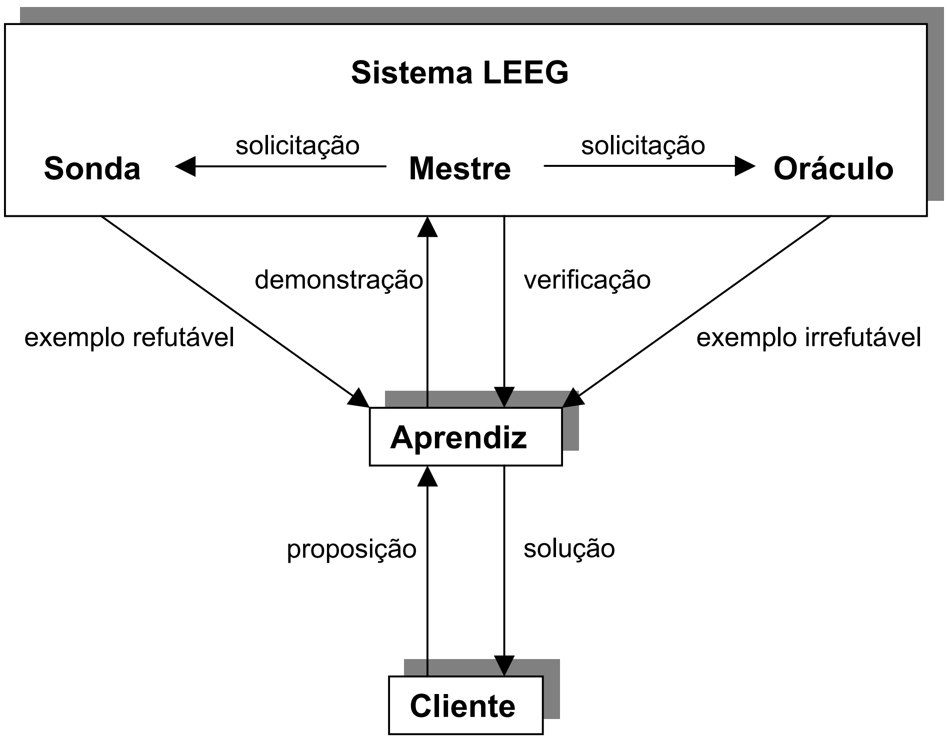

FIGURA 3.5 – Protocolo MOSCA no LEEG

3.6 Considerações Finais

Este capítulo descreveu o protocolo de aprendizagem MOSCA e suas aplicações em sistemas já devolvidos anteriormente.

Apresentou-se a proposta de desenvolvimento do sistema LEEG. O objetivo do sistema LEEG é promover uma aprendizagem construtiva, baseada em exemplos e contra-exemplos, que servem como dicas para auxiliar o Aprendiz na construção das demonstrações da Geometria Euclidiana Plana.

Para a validação deste trabalho, vai-se considerar o agente Aprendiz humano, ficando para trabalho futuro a implementação do agente Aprendiz artificial.

Como resultado do trabalho desenvolvido neste capítulo, obteve-se a publicação do artigo *"Aprendiz's Learning in Geometry Demonstration"* [NOT 01].

No capítulo que segue, a especificação do sistema LEEG está desenvolvida, detalhando a estrutura de dados do sistema, além do comportamento e do relacionamento entre os agentes e das mensagens enviadas pelos agentes Oráculo e Sonda ao agente Aprendiz.

4 Especificação do Sistema LEEG

Este capítulo está relacionado com os artigos *"Hyper-Automaton System Applied to Geometry Demonstration Environment"* [MAC 01] e *"Knowledge Anticipation on Agents Relationship in the Geometry Demonstration Proof Environment"* [NOT 01a] e apresenta a especificação de um protótipo do sistema LEEG, desenvolvido com a finalidade de ilustrar e validar a proposta deste trabalho.

É detalhada a estrutura de dados do LEEG, que é composta por uma base de conhecimento, uma base de exemplos, subdividida em um módulo de exemplos irrefutáveis e um módulo de exemplos refutáveis, e um conjunto de proposições.

Também são estudadas e estabelecidas nesse capítulo, todas as interações possíveis entre os agentes do sistema.

4.1 Descrição

O sistema LEEG é um sistema para a aprendizagem de demonstrações da Geometria Euclidiana Plana, proposto com o objetivo de auxiliar o agente Aprendiz no processo de aprendizagem das demonstrações geométricas.

Na concepção deste primeiro protótipo, optou-se por adotar o agente Aprendiz humano, apresentado no capítulo 3 (seção 3.5.2.2), já que esta abordagem permite analisar o funcionamento do sistema e as interações entre todos os agentes que o compõem, durante o processo de construção das demonstrações, não comprometendo a validação e avaliação da proposta deste trabalho.

Os agentes Mestre, Oráculo e Sonda são artificiais e a descrição de seus comportamentos também podem ser conferidas no capítulo 3.

As demonstrações dedutivas estão desenvolvidas na interface do LEEG pela forma textual (capítulo 2, seção 2.1.2). Porém, por questões práticas, optou-se em organizar as demonstrações em uma tabela constituída basicamente de duas colunas nomeadas *Enunciado* e *Regra de Dedução*, nas quais o agente Aprendiz deverá desenvolver sua demonstração. Para cada campo da coluna de enunciados, terá um campo equivalente na coluna de regra de dedução, que deverá ser rigorosamente preenchido pela regra de dedução que conclui o respectivo enunciado. Somente os enunciados que não são deduzidos de regras de dedução, considerados nesse trabalho apenas como sendo as hipóteses, podem ter este campo não preenchido.

As regras de dedução aceitas pelo LEEG são as 23 definições, os 5 axiomas e os 5 postulados da Geometria Plana de Euclides (encontrados no capítulo 2, seção 2.1.4), aplicados aos elementos corretos. Quando se fala em aplicar a regra de dedução aos elementos corretos, entende-se o seguinte: tomando-se como exemplo de regra de dedução o Postulado 1, que diz "*É possível construir uma linha reta a partir de qualquer ponto para qualquer ponto*", observa-se que ela refere-se à construção de um segmento a partir de dois pontos dados, o que significa que ela deve ser aplicada a exatamente dois pontos.

Entretanto, para padronizar o funcionamento do sistema, foi preciso incluir uma nova regra de dedução, que não constitui nenhum axioma, definição ou postulado. Essa nova regra é a operação de *interseção*, utilizada em grande parte das demonstrações para a dedução de novos pontos necessários para o prosseguimento das mesmas.

A estrutura de uma demonstração também pode ser um grafo (capítulo 2, seção 2.1.2), onde todos os enunciados da Geometria utilizados para a prova encontram-se organizados hierarquicamente e as arestas de ligação entre os enunciados são os termos que justificam a utilização dos mesmos (regras de dedução). O primeiro e último enunciados da demonstração devem ser, respectivamente, a hipótese e a tese da proposição.

Para um melhor esclarecimento sobre o desenvolvimento de demonstrações dedutivas, apresenta-se a seguir a demonstração das Proposições 1 e 2 (que foram implementadas no protótipo), nas representações textual (tab.4.1 e tab.4.2) e de grafo (fig.4.1 e fig.4.2). A representação textual prevê uma das formas de construção da demonstração realizada pelo Aprendiz na interface do sistema e a representação de grafo foi utilizada para gerar os autômatos que estruturam e modelam a base de conhecimento do sistema (seção 4.2.1).

Proposição 1: C*onstruir um triângulo equilátero sobre um segmento de reta dado*.

A partir do enunciado da proposição, devem ser identificadas a hipótese e a tese. Neste caso, a hipótese da proposição é o *segmento de reta* e a tese é o *triângulo equilátero*.

Para facilitar o entendimento e acompanhamento da demonstração, é usual nomear os elementos geométricos que estão sendo utilizados na construção.

TABELA 4.1 – Representação textual da demonstração da Proposição 1

Enunciado	Regra de Dedução
segmento AB	(Hipótese)
círculo A-AB	Postulado 3 (A, AB)
círculo B-AB	Postulado 3 (B, AB)
ponto C	Interseção (A-AB, B-AB)
segmento AC	Postulado 1 (A, C)
segmento BC	Postulado 1 (B, C)
segmento AC = segmento AB	Definição 15 (AC, AB)
segmento BC = segmento AB	Definição 15 (BC, AB)
segmento AC = segmento BC	Axioma 1 (AB, AC, BC)
triângulo ABC = equilátero	Definição 20 (AB, AC, BC)

O campo da coluna de regras de dedução referente à hipótese da proposição não precisa ser incluído na demonstração que está sendo elaborada pelo Aprendiz, uma vez que não é um enunciado deduzido a partir de uma regra de dedução e sim extraído da proposição.

Cada passo do raciocínio do Aprendiz, ou seja, cada linha da tabela de demonstração, é supervisionado pelo Mestre. O passo seguinte só será habilitado se o raciocínio corrente estiver correto. Caso contrário, os agentes Oráculo e Sonda são sinalizados para enviar exemplos e contra-exemplos que alertem ao Aprendiz de seu erro e permitam sua correção. Este assunto está detalhado na seção 4.2.2.

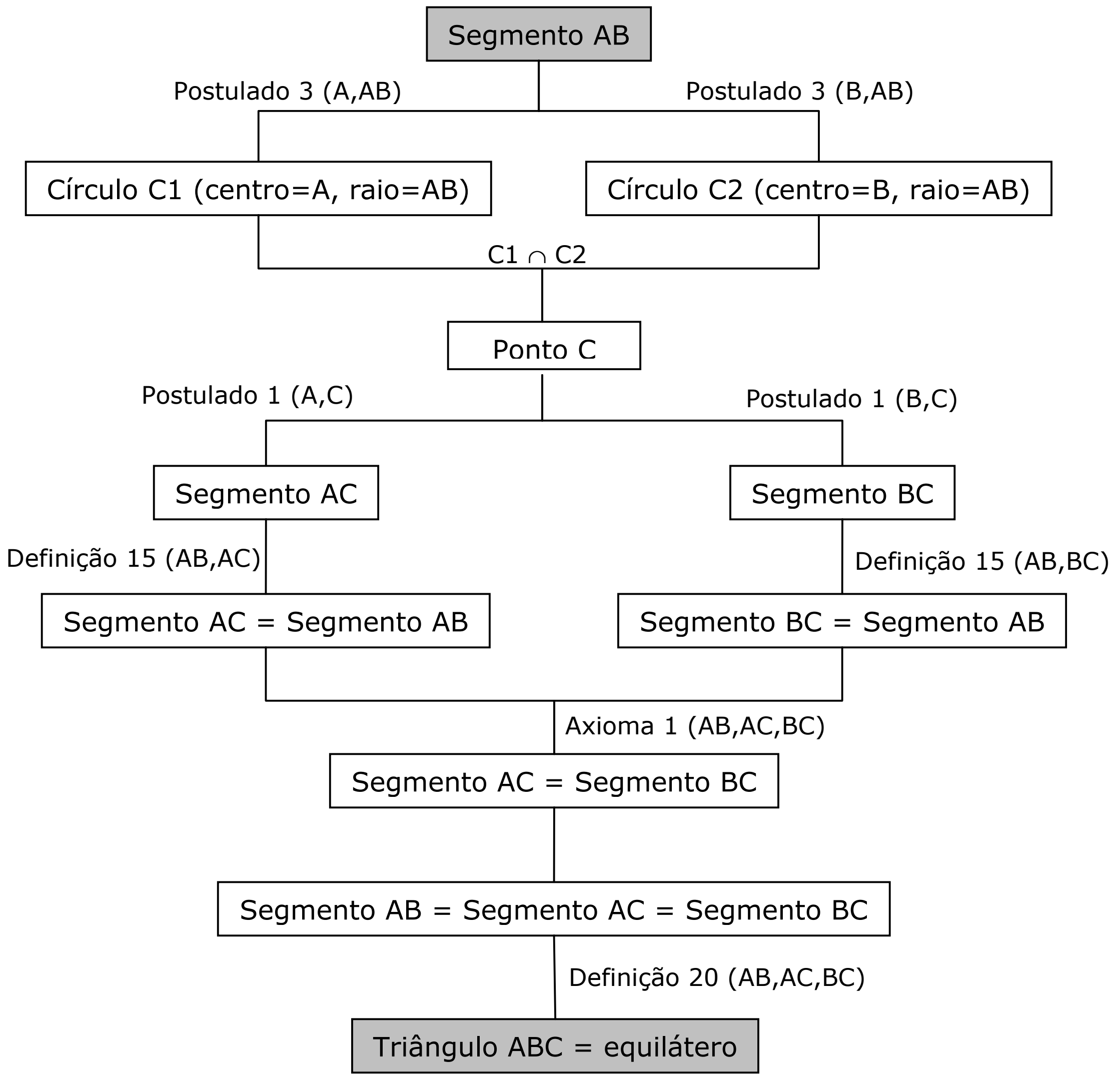

FIGURA 4.1 - Representação em grafo da demonstração da Proposição 1

Os nodos destacados (primeiro e último) correspondem, respectivamente, à hipótese e à tese da proposição, que devem ser identificados no seu enunciado. Para a demonstração da proposição, partiu-se da hipótese (assumida como verdadeira) e, através de uma seqüência lógica de enunciados, provou-se ser verdadeira a tese. Note-se que cada enunciado lançado é rigorosamente justificado pelos axiomas, definições ou postulados. Os enunciados que não apresentam justificativa são justificados por enunciados anteriores da própria demonstração. Os enunciados que ocorrem simultaneamente no grafo possuem a mesma hierarquia. A ausência de um enunciado ou ligação representa uma inconsistência na demonstração.

Proposição 2: *Construir um segmento de reta igual a um segmento de reta dado com uma das extremidades em ponto dado.*

Pelo enunciado, identifica-se que a hipótese da proposição são o *segmento de reta* e o *ponto* dados e a tese é um *novo segmento de reta*, de mesma medida que o primeiro, construído com extremidade sobre o ponto dado.

Percebe-se que a Proposição 2 possui duas hipóteses, o que faz lembrar que a hipótese de um teorema pode ser uma única afirmação ou um conjunto delas.

TABELA 4.2 - Representação textual da demonstração da Proposição 2

Enunciado	Regra de Dedução
ponto A	(Hipótese)
segmento BC	(Hipótese)
segmento AB	Postulado 1 (A, B)
triângulo ABD	Proposição 1 (AB)
círculo B-BC	Postulado 3 (B, BC)
reta AE	Postulado 2 (DA)
reta BF	Postulado 2 (DB)
ponto G	Interseção (B-BC, BF)
segmento DG	Postulado 1 (D, G)
segmento BG	Postulado 1 (B, G)
círculo D-DG	Postulado 3 (D, DG)
ponto L	Interseção (D-DG, AE)
segmento AL	Postulado 1 (A, L)
segmento DL	Postulado 1 (D, L)
segmento DG = segmento DL	Definição 15 (DG, DL)
segmento DB = segmento DA	Definição 20 (DB, DA, BA)
segmento BG = segmento AL	Axioma 3 (BG, AL)
segmento BC = segmento BG	Definição 15 (BC, BG)
segmento AL = segmento BC	Axioma 1 (BG, BC, AL)

Observa-se que os enunciados envolvidos nas demonstrações em ambas representações são os mesmos, mudando realmente apenas a estrutura dos mesmos, o que mostra a equivalência entre as duas representações.

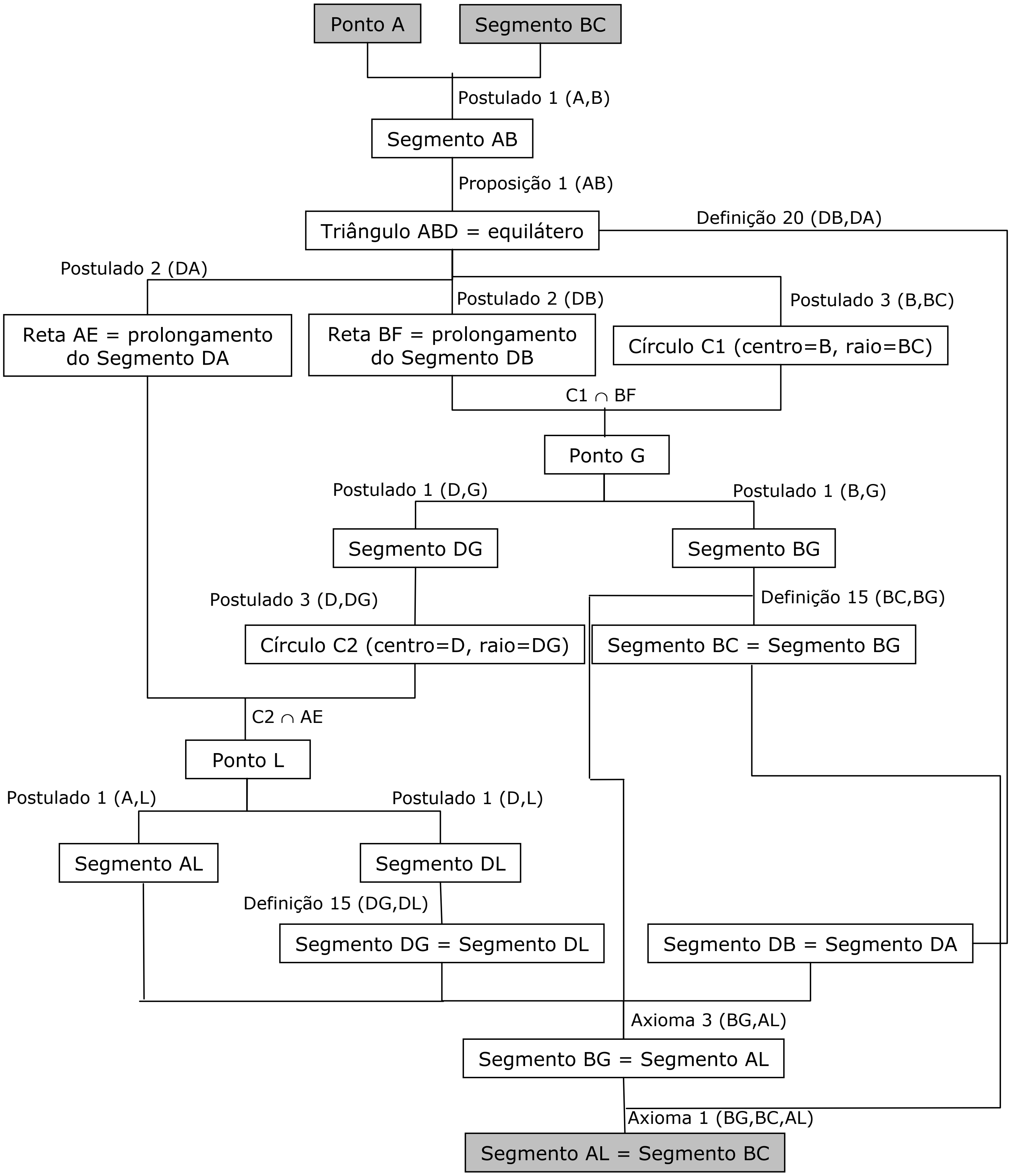

FIGURA 4.2 - Representação em grafo da demonstração da Proposição 2

4.2 Estrutura de Dados

Os dados do sistema estão estruturados da seguinte forma:

- *Base de Conhecimento*, constituída pelas definições, axiomas, postulados e proposições já demonstradas;
- *Base de Proposições* que são submetidas à demonstração;
- *Base de Exemplos*, dividida em dois módulos:
  - *Módulo de Exemplos Irrefutáveis*
  - *Módulo de Exemplos Refutáveis*

A fig.4.3 mostra esta estrutura de dados do sistema.

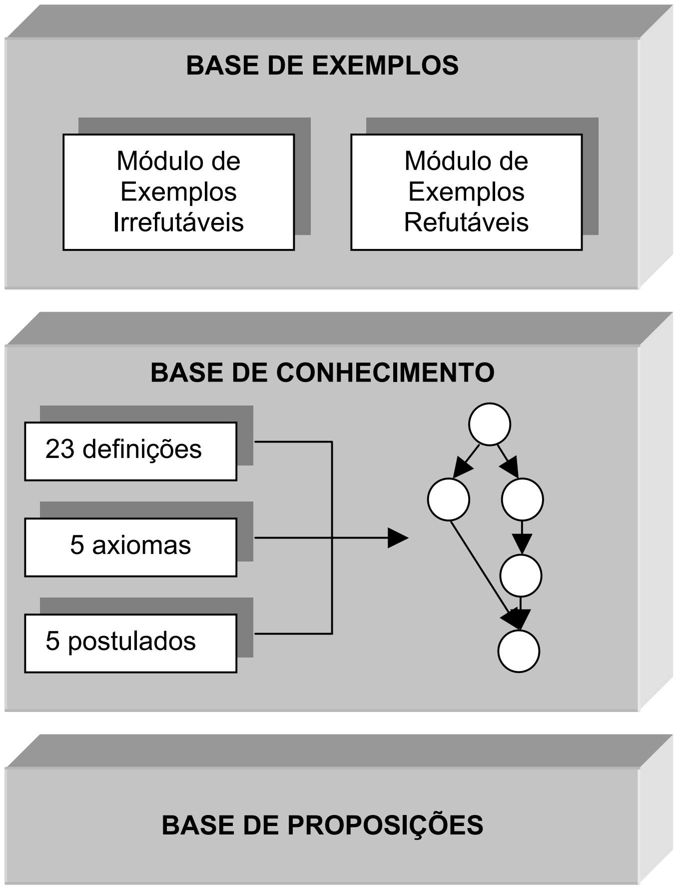

FIGURA 4.3 - Estrutura dos dados do sistema LEEG

4.2.1 Base de Conhecimento do Sistema

A base de conhecimento do sistema é o conjunto completo de todos os enunciados possíveis que poderão ser utilizados pelo Aprendiz na demonstração de cada uma das proposições. As demonstrações das proposições estão desenvolvidas na base do sistema, obedecendo à seqüência lógica dos enunciados e justificativas que deve ser reproduzida pelo Aprendiz. No caso do sistema LEEG, a base de conhecimento pode ser composta por 23 definições, 5 axiomas, 5 postulados e 48 teses de proposições, já demonstradas.

Na base de conhecimento do sistema LEEG, esta estrutura das demonstrações está organizada e implementada através da utilização do modelo de Autômato Finito, que permite uma organização dos dados de forma bastante eficiente. A aplicação do modelo de Autômato Finito na estrutura de demonstrações dedutivas justifica-se pela:

- facilidade na representação do conhecimento;
- possibilidade de verificação da aprendizagem pela comparação de sub-estruturas dos autômatos;
- possibilidade de observar redundância de computação, o que, na verdade, modela as diferentes formas de raciocínio. Sendo assim, pode-se estudar o raciocínio do agente Aprendiz na diversificação em demonstrar as proposições.

Como já estudado no capítulo 2 (seção 2.3), um autômato é representado por uma 5-upla $M=(\Sigma, Q, \delta, q_0, F)$, onde: Σ é o *alfabeto de símbolos de entrada*; Q é o *conjunto de estados* possíveis do autômato, o qual é finito; δ é a *função programa* ou *função de transição*, a qual é uma função parcial; q_0 é o *estado inicial* do autômato, tal que q_0 é elemento de Q; e F é o *conjunto de estados finais*, tal que F está contido em Q.

O autômato representa todos os possíveis caminhos de dedução que levam da hipótese à tese correta. Isso significa que a linguagem aceita pelo autômato é a dedução correta do sistema. O conjunto de axiomas, postulados e definições, além dos enunciados utilizados para justificar a passagem de um estado a outro da demonstração, é visto como o alfabeto Σ de símbolos de entrada pelo autômato. O conjunto Q de estados do autômato representa o estado corrente do processo de raciocínio da demonstração: os objetos intermediários que são utilizados na construção da tese. A função de transição δ é representada pelos elos de ligação dos enunciados, isto é, por cada justificativa que permite a transição de um enunciado a outro. Logo, a função de transição é os axiomas, postulados e definições aplicados a determinados elementos (ponto, segmento, etc.), além de proposições já demonstradas ou estados anteriores da demonstração corrente. O estado inicial q_0 corresponde à hipótese da proposição que está sendo proposta ao Aprendiz, e o estado final q_f corresponde à tese da mesma.

As fig.4.4 e fig.4.5 mostram o grafo das demonstrações das Proposições 1 e 2 modeladas por Autômatos Finitos.

As funções de transição dos autômatos estão representadas nas tab.4.3 e tab.4.4.

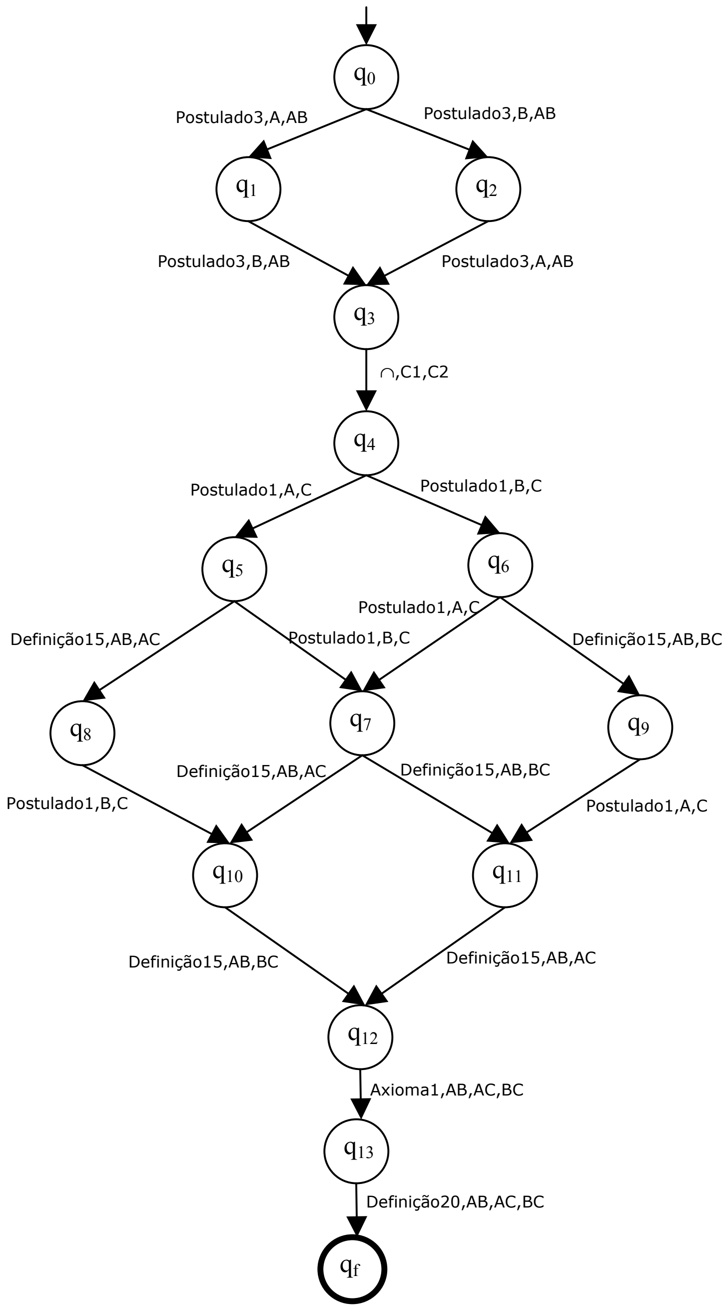

FIGURA 4.4 - Autômato da demonstração da Proposição1

Percebe-se que a demonstração pode ser desenvolvida por diferentes raciocínios, ou seja, diversos caminhos levam ao estado final do autômato. Isso porque os enunciados que ocorrem simultaneamente na demonstração com estrutura de grafo foram representados por todas as possíveis ordenações seqüenciais no autômato. Essa modelagem permite considerar diferentes raciocínios produzidos pelo Aprendiz no processo de aprendizagem da demonstração, que são equivalentes e aceitos como corretos pelo sistema.

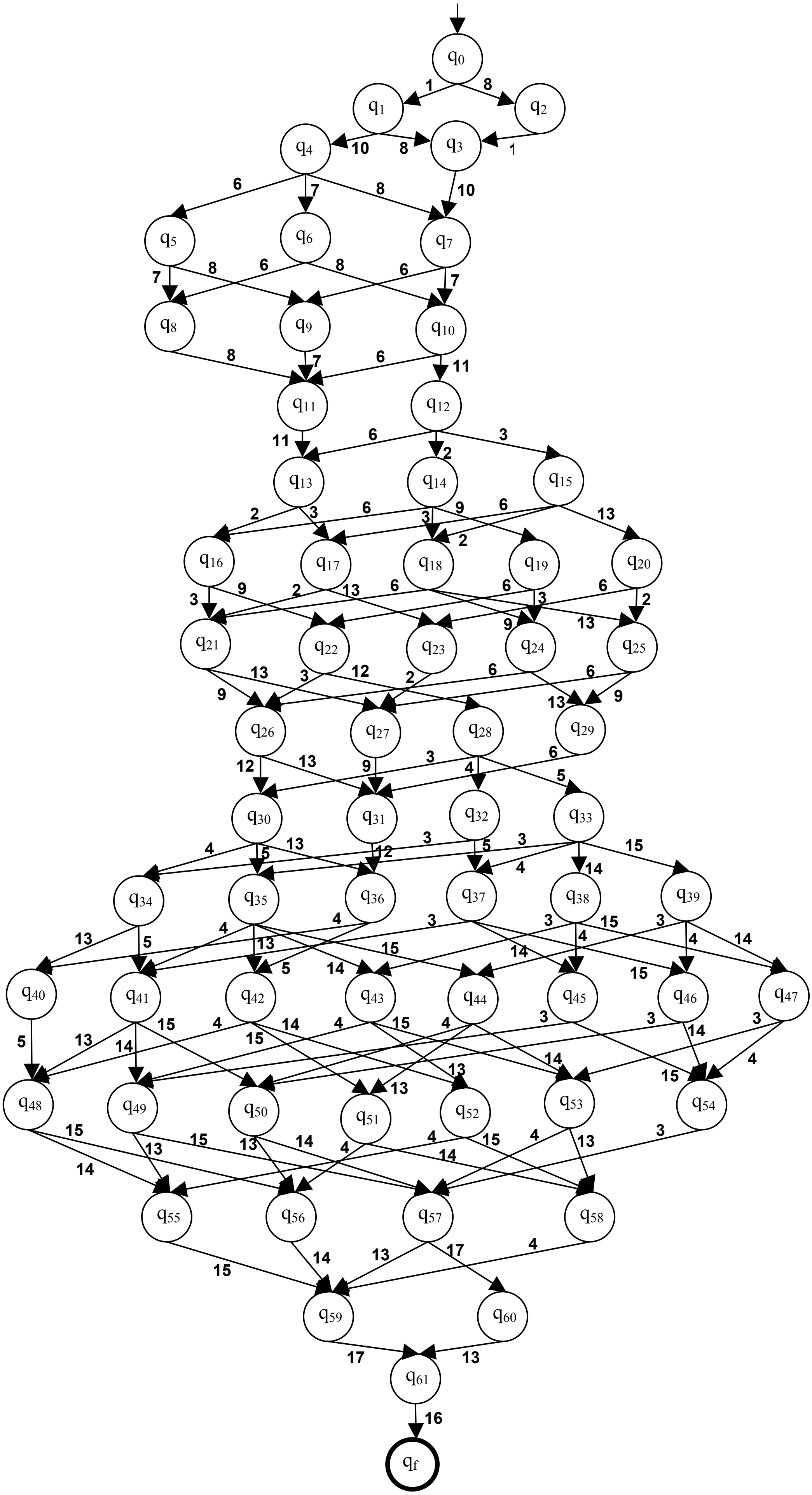

FIGURA 4.5 - Autômato da demonstração da Proposição 2

TABELA 4.3 - Função de transição do autômato 1

δ_1	**P3,A,AB**	**P3,B,AB**	**∩,C1,C2**	**P1,A,C**	**P1,B,C**	**D15,AB,AC**	**D15,AB,BC**	**A1,AB,AC,BC**	**D20,AB,AC,BC**
q_0	q_1	q_2							
q_1		q_3							
q_2	q_3								
q_3			q_4						
q_4				q_5	q_6				
q_5					q_7	q_8			
q_6				q_7			q_9		
q_7						q_{10}	q_{11}		
q_8					q_{10}				
q_9				q_{11}					
q_{10}							q_{12}		
q_{11}						q_{12}			
q_{12}								q_{13}	
q_{13}									q_f
q_f									

TABELA 4.4 - Função de transição do autômato 2

δ_2	**P1,A,B**	**P1,D,G**	**P1,B,G**	**P1,A,L**	**P1,D,L**	**P2,DA**	**P2,DB**	**P3,B,BC**	**P3,D,DG**	**Pr1,AB**	**∩,C1,BF**	**∩,C2,AE**	**D15,BC,BG**	**D15,DG,DL**	**D20,DB,DA,BA**	**A1,BG,BC,AL**	**A3,BG,AL**
q_0	q_1							q_2									
q_1								q_3		q_4							
q_2	q_3																
q_3										q_7							
q_4						q_5	q_6	q_7									
q_5							q_8	q_9									
q_6						q_8		q_{10}									
q_7						q_9	q_{10}										
q_8								q_{11}									
q_9							q_{11}										
q_{10}						q_{11}					q_{12}						
q_{11}											q_{13}						
q_{12}		q_{14}	q_{15}			q_{13}											
q_{13}		q_{16}	q_{17}														
q_{14}			q_{18}			q_{16}			q_{19}								
q_{15}		q_{18}				q_{17}							q_{20}				
q_{16}			q_{21}						q_{22}								
q_{17}		q_{21}											q_{23}				
q_{18}						q_{21}			q_{24}				q_{25}				
q_{19}			q_{24}			q_{22}											
q_{20}		q_{25}				q_{23}											
q_{21}									q_{26}				q_{27}				

q_{22}			q_{26}									q_{28}					
q_{23}		q_{27}															
q_{24}						q_{26}							q_{29}				
q_{25}						q_{27}			q_{29}								
q_{26}												q_{30}	q_{31}				
q_{27}									q_{31}								
q_{28}			q_{30}	q_{32}	q_{33}												
q_{29}						q_{31}											
q_{30}				q_{34}	q_{35}								q_{36}				
q_{31}												q_{36}					
q_{32}			q_{34}		q_{37}												
q_{33}			q_{35}	q_{37}										q_{38}	q_{39}		
q_{34}					q_{41}								q_{40}				
q_{35}				q_{41}									q_{42}	q_{43}	q_{44}		
q_{36}				q_{40}	q_{42}												
q_{37}			q_{41}											q_{45}	q_{46}		
q_{38}			q_{43}	q_{45}											q_{47}		
q_{39}			q_{44}	q_{46}										q_{47}			
q_{40}					q_{48}												
q_{41}													q_{48}	q_{49}	q_{50}		
q_{42}				q_{48}										q_{52}	q_{51}		
q_{43}				q_{49}									q_{52}		q_{53}		
q_{44}				q_{50}									q_{51}	q_{53}			
q_{45}			q_{49}												q_{54}		
q_{46}			q_{50}											q_{54}			
q_{47}			q_{53}	q_{54}													
q_{48}														q_{55}	q_{56}		
q_{49}													q_{55}		q_{57}		
q_{50}													q_{56}	q_{57}			
q_{51}				q_{56}										q_{58}			
q_{52}				q_{55}											q_{58}		
q_{53}				q_{57}									q_{58}				
q_{54}			q_{57}														
q_{55}															q_{59}		
q_{56}														q_{59}			
q_{57}													q_{59}				q_{60}
q_{58}				q_{59}													
q_{59}																	q_{61}
q_{60}													q_{61}				
q_{61}																q_f	
q_f																	

As transições do Autômato 2 estão listadas na tab.4.5, pois, por motivo de espaço físico, não foi possível apresentá-las no grafo.

TABELA 4.5 – Transições do Autômato 2

Número da Transição	Função de Transição
1	Postulado 1 (A, B)
2	Postulado 1 (D, G)
3	Postulado 1 (B, G)
4	Postulado 1 (A, L)
5	Postulado 1 (D, L)
6	Postulado 2 (DA)
7	Postulado 2 (DB)
8	Postulado 3 (B, BC)
9	Postulado 3 (D, DG)
10	Proposição 1 (AB)
11	Interseção (B-BC, BF)
12	Interseção (D-DG, AE)
13	Definição 15 (BC, BG)
14	Definição 15 (DG, DL)
15	Definição 20 (DB, DA, BA)
16	Axioma 1 (BG, BC, AL)
17	Axioma 3 (BG, AL)

4.2.2 Base de Exemplos

A base de exemplos é formada pelo conjunto de intervenções enviadas pelo sistema (através dos agentes Oráculo e Sonda) ao Aprendiz na forma de mensagem. Essas intervenções têm duas finalidades principais:

- emissão de exemplos que contenham *dicas* incontestáveis de regras de dedução, a fim de conduzir o agente Aprendiz à construção correta da demonstração, ou seja, possibilitar a inclusão de regras de dedução coerentes com a demonstração corrente;
- emissão de contra-exemplos, sempre que o Aprendiz cometer um erro de enunciado, a fim de alertá-lo sobre o erro cometido e auxiliar na sua correção.

Os erros cometidos pelo Aprendiz podem ser referentes aos enunciados ou às regras de dedução. A *incoerência* de uma demonstração é reflexo de enunciados deduzidos a partir de regras de dedução incorretas e a *inconsistência* de uma demonstração é a introdução de enunciados sem a aplicação de uma regra de dedução.

Vale lembrar que o agente responsável pela decisão de emissão de mensagens é o Mestre, que acompanha e verifica, passo a passo, o raciocínio que está sendo desenvolvido pelo Aprendiz. É mediante a solicitação dele que os agentes Oráculo e Sonda conduzem o Aprendiz à construção correta.

Dessa forma, o Oráculo auxilia o Aprendiz, mediante sinalização do Mestre, com a emissão de exemplos que encaminhem a concluir a regra de dedução correta. Por outro lado, o Sonda é sinalizado pelo Mestre sempre que o Aprendiz comete um erro de enunciado na demonstração, para que este envie um contra-exemplo que alerte ao Aprendiz de seu erro e possibilite reflexão e sua correção.

As intervenções dos agentes Oráculo e Sonda no raciocínio do Aprendiz estão organizadas da seguinte maneira:

- para cada enunciado errado incluído pelo Aprendiz na demonstração, o agente Sonda é acionado para enviar mensagens de contra-exemplo, até que o Aprendiz identifique o erro e o corrija;
- para cada regra de dedução errada incluída pelo Aprendiz na demonstração, o agente Oráculo é acionado para enviar dicas que permitam a correção do erro;
- para cada dupla enunciado/regra de dedução correta incluída pelo Aprendiz e verificada pelo Mestre, o agente Oráculo é acionado para enviar dicas referentes ao próximo estado da demonstração[1]. Entretanto, estas dicas não são enviadas sempre que o Aprendiz acerta um estado de demonstração. São acionadas somente nas situações em que determinado tempo decorre, estabelecido neste primeiro protótipo em 10 minutos, e nenhuma ação do Aprendiz é identificada pelo Mestre.

As subseções seguintes explicam com mais detalhes as mensagens de intervenção do LEEG, tratando dos dois módulos que compõem a Base de Exemplos: Módulo de Exemplos Irrefutáveis e Módulo de Exemplos Refutáveis.

4.2.2.1 Módulo de Exemplos Irrefutáveis

O Módulo de Exemplos Irrefutáveis é acessado apenas pelo agente Oráculo, responsável pelo envio de mensagens de exemplos e dicas. As mensagens contidas neste módulo são irrefutáveis, isto é, corretas e podem ser aceitas pelo Aprendiz com total confiança.

As mensagens deste módulo estão divididas em duas categorias, identificadas por suas funcionalidades. São elas: *Mensagens de Correção* e *Mensagens de Incentivo*.

As mensagens de correção são disparadas mediante identificação de erro nas regras de dedução do Aprendiz e têm como objetivo fornecer dicas que alertem o Aprendiz sobre o erro e encaminhem sua correção. A fig.4.6 mostra a interação entre os agentes Aprendiz, Mestre e Oráculo durante o processo de identificação do erro e produção da mensagem de correção.

---

[1] Entende-se por estado da demonstração, no LEEG, cada linha da tabela de demonstração, ou seja, cada dupla enunciado/regra de dedução.

FIGURA 4.6 – Produção das mensagens de correção

Em suma, o Aprendiz produz uma regra de dedução errada, o Mestre identifica o erro através da comparação com a base de conhecimento do LEEG e sinaliza o Oráculo para que envie ao Aprendiz uma mensagem de correção relativa ao estado da demonstração em que o Aprendiz se encontra.

Estas mensagens ainda podem ser classificadas em três níveis distintos, de acordo com erro identificado:

- *dica sobre o tipo de elemento que deve ser aplicada a regra de dedução* - enviada quando detectado que o Aprendiz comete erro relacionado ao tipo de elemento geométrico (ponto, segmento, ...) no qual a regra de dedução deve ser aplicada. Ou seja, o Aprendiz acerta a regra de dedução, mas não acerta o tipo de elemento de aplicação. (Exemplo de mensagem: *"O Postulado 3 deve ser aplicado a um ponto e um segmento."*)
- *dica que especifica os elementos nos quais deve ser aplicada a regra de dedução* - enviada quando detectado que o Aprendiz comete erro apenas no(s) elemento(s) (e não no tipo de elemento). (Exemplo de mensagem: *"Atenção: Observe os elementos que você utilizou no enunciado!")*
- *mensagem advertindo incoerência entre enunciado e regra de dedução* - enviada para alertar ao Aprendiz que a regra de dedução inserida é incoerente com o respectivo enunciado, ou seja, o enunciado não pode ser deduzido da referida regra. (Mensagem: *"A regra de dedução não está coerente com o enunciado.")*

Por outro lado, as mensagens de incentivo são disparadas mediante a constatação de dois fatos simultâneos: enunciado e respectiva regra de dedução corretos e ausência de ação do Aprendiz por 10 minutos, que está sendo interpretado por indecisão ou abandono da aprendizagem da demonstração. Essas mensagens têm o objetivo de incentivar o Aprendiz a prosseguir com a demonstração quando se percebe indecisão ou abandono, apresentando dicas relativas ao próximo estado da demonstração. A fig.4.7 apresenta a interação entre os agente Aprendiz, Mestre e Oráculo durante o processo de produção das mensagens de incentivo.

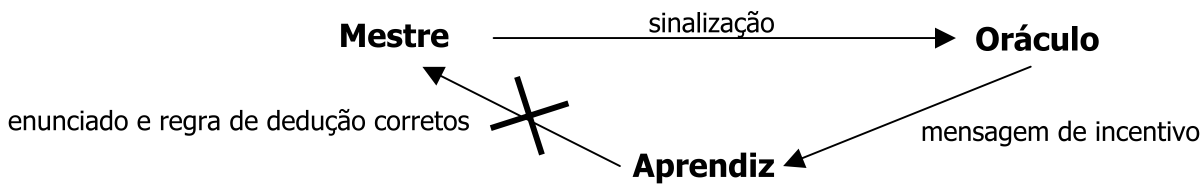

FIGURA 4.7 – Produção das mensagens de incentivo

Cada estado das demonstrações está associado a uma mensagem de incentivo. As mensagens de incentivo das proposições 1 e 2 estão apresentadas nas tab.4.6 e tab.4.7, respectivamente.

TABELA 4.6 – Mensagens de incentivo da Proposição 1

Estado	Saída
q_0	Pense na construção com régua e compasso: o círculo nos garante a construção de segmentos iguais.
q_1	Ainda falta construir um círculo...
q_2	Ainda falta construir um círculo...
q_3	O terceiro vértice do triângulo vem da interseção dos dois círculos.
q_4	Atenção! Os lados do triângulo já podem ser construídos.
q_5	Por que você não constrói o terceiro lado do triângulo?
q_6	Por que você não constrói o terceiro lado do triângulo?
q_7	Por que os círculos foram construídos? ...para garantir a igualdade dos lados...
q_8	Por que você não constrói o terceiro lado do triângulo?
q_9	Por que você não constrói o terceiro lado do triângulo?
q_{10}	Por que os círculos foram construídos? ...para garantir a igualdade dos lados...
q_{11}	Por que os círculos foram construídos? ...para garantir a igualdade dos lados...
q_{12}	É preciso provar que os três lados do triângulo são iguais entre si.
q_{13}	Você já tem todas as deduções necessárias para concluir a tese!
q_f	

TABELA 4.7 – Mensagens de incentivo da Proposição 2

Estado	Saída
q_0	Que tal começar construindo o segmento &0&1?
q_1	Ajudaria se você construísse um triângulo equilátero sobre o segmento &0&1.
q_2	Que tal construir o segmento &0&1?
q_3	Ajudaria se você construísse um triângulo equilátero sobre o segmento &0&1.
q_4	O círculo é sempre indicado quando queremos construir segmentos iguais.
q_5	Prolongue o outro lado do triângulo.
q_6	Prolongue o outro lado do triângulo.
q_7	Prolongue os lados &3&0 e &3&1.
q_8	O círculo é sempre indicado quando queremos construir segmentos iguais.
q_9	Prolongue o outro lado do triângulo.
q_{10}	Prolongue o outro lado do triângulo.
q_{11}	Se você não fizer a interseção do círculo com a reta, não vai conseguir ir adiante!
q_{12}	Prolongue o outro lado do triângulo.

q_{13}	Você precisa construir o segmento &3&6.
q_{14}	Prolongue o outro lado do triângulo.
q_{15}	Prolongue o outro lado do triângulo.
q_{16}	Construa o segmento &1&6.
q_{17}	Você precisa construir o segmento &3&6.
q_{18}	Prolongue o outro lado do triângulo.
q_{19}	Prolongue o outro lado do triângulo.
q_{20}	Prolongue o outro lado do triângulo.
q_{21}	Lembre-se: o círculo é sempre indicado quando queremos construir segmentos iguais.
q_{22}	Se você não fizer a interseção do segundo círculo com a reta, não vai conseguir ir adiante!
q_{23}	Você precisa construir o segmento &3&6 para prosseguir.
q_{24}	Prolongue o outro lado do triângulo.
q_{25}	Prolongue o outro lado do triângulo.
q_{26}	Se você não fizer a interseção do segundo círculo com a reta, não vai conseguir ir adiante!
q_{27}	Lembre-se: o círculo é sempre indicado quando queremos construir segmentos iguais.
q_{28}	Construa o segmento &3&7.
q_{29}	Prolongue o outro lado do triângulo.
q_{30}	Construa o segmento &3&7.
q_{31}	Se você não fizer a interseção do segundo círculo com a reta, não vai conseguir ir adiante!
q_{32}	Construa o segmento &3&7.
q_{33}	Construa o segmento &0&7.
q_{34}	Construa o segmento &3&7.
q_{35}	Construa o segmento &0&7.
q_{36}	Construa o segmento &3&7.
q_{37}	Construa o segmento &1&6.
q_{38}	Construa o segmento &0&7.
q_{39}	Construa o segmento &0&7.
q_{40}	Você precisa construir o segmento &3&7.
q_{41}	Você já pode começar a deduzir as igualdades entre os segmentos.
q_{42}	Construa o segmento &0&7.
q_{43}	Construa o segmento &0&7.
q_{44}	Construa o segmento &0&7.
q_{45}	Construa o segmento &1&6.
q_{46}	Construa o segmento &1&6.
q_{47}	Construa o segmento &0&7.
q_{48}	Continue deduzindo as igualdades entre os segmentos.
q_{49}	Continue deduzindo as igualdades entre os segmentos.
q_{50}	Continue deduzindo as igualdades entre os segmentos.
q_{51}	Construa o segmento &0&7.
q_{52}	Construa o segmento &0&7.
q_{53}	Construa o segmento &0&7.
q_{54}	Você precisa construir o segmento &1&6 para prosseguir.
q_{55}	Continue deduzindo as igualdades entre os segmentos.
q_{56}	Continue deduzindo as igualdades entre os segmentos.
q_{57}	Continue deduzindo as igualdades entre os segmentos.

q_{58}	Você precisa construir o segmento &0&7 para prosseguir.
q_{59}	Continue deduzindo as igualdades entre os segmentos.
q_{60}	Continue deduzindo as igualdades entre os segmentos.
q_{61}	Você já tem todas as deduções necessárias para concluir a tese!
q_f	

4.2.2.2 Módulo de Exemplos Refutáveis

O Módulo de Exemplos Refutáveis é acessado apenas pelo agente Sonda, responsável pelo envio de contra-exemplos, que podem ser refutados pelo Aprendiz.

As mensagens desse módulo, chamadas de *Mensagens de Reflexão*, são disparadas apenas nas situações em que Aprendiz comete erros de enunciado. Elas têm como objetivo alertá-lo sobre o erro cometido, criando situações que provoquem incerteza, para que o Aprendiz reflita sobre o enunciado concluído. A fig.4.8 apresenta a interação entre os agentes Aprendiz, Mestre e Sonda durante o processo de produção das mensagens de reflexão.

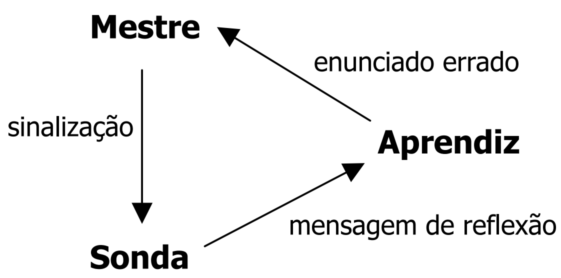

FIGURA 4.8 – Produção das mensagens de reflexão

Essas mensagens ainda estão classificadas em quatro níveis distintos, de acordo com o erro cometido pelo Aprendiz. São eles:

- utilização de elementos (pontos, segmentos, ...) que não foram construídos anteriormente, ou seja, que não existem. (Exemplo de mensagem: *"O que é segmento BC?"*)
- utilização de variáveis incorretas, ou seja, construir, por exemplo, um segmento com a utilização de três pontos, ou construir um triângulo com a utilização de um ponto, etc. (Exemplo de mensagem: *"Pense...Quantos vértices determinam um triângulo?"*)
- inserir enunciados corretos, mas logicamente desordenados. (Mensagem: *"Você ainda não pode concluir isto!"*)
- qualquer outra situação não prevista nas situações acima. (Mensagem: *"Enunciado incorreto."*)

4.2.3 Base de Proposições

A base de proposições, ou banco das possíveis proposições a serem submetidas ao Aprendiz, é determinada pelo escopo da base de conhecimento. No caso da Geometria Euclidiana Plana, existem 48 possíveis proposições a serem submetidas pelo Cliente ao Aprendiz. O agente responsável pelo acesso a esse conjunto de proposições é o Cliente, que deve escolher uma proposição e submetê-la ao Aprendiz para ser provada. A forma de escolha da proposição pode ser *estruturada* ou *randômica*.

No caso estruturado - método adotado neste trabalho - a ordem de dependência entre proposições é respeitada, isto é, uma proposição só estará disponibilizada para submissão se as demais proposições necessárias para sua demonstração já foram efetivamente provadas pelo Aprendiz. Dessa forma, a aquisição do conhecimento é construída passo a passo. Por esse fato, é chamada de *indutiva*. A relação de dependência entre as proposições inseridas no protótipo do LEEG está indicada na fig.4.9.

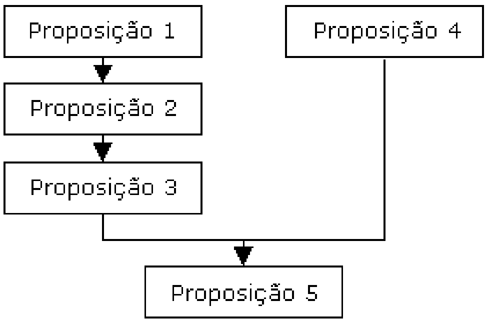

FIGURA 4.9 – Dependências entre as proposições disponíveis no LEEG

Dessa forma, ao iniciar o sistema, apenas as proposições 1 e 4 estão disponibilizadas para demonstração, já que não dependem de nenhuma outra proposição para sua prova. Assim que a Proposição 1 é provada, a Proposição 2 passa a estar disponibilizada para submissão ao Aprendiz, e assim por diante.

No caso randômico, não há ordem entre as submissões, ou seja, ao iniciar o sistema, todas as proposições estão disponíveis para submissão ao Aprendiz, mas suas relações de dependência são conservadas. Isto significa que, se for submetida ao Aprendiz uma proposição cuja demonstração exige o resultado de outra(s) proposição(s), este deve demonstrá-la(s) para, depois, prosseguir com a demonstração solicitada. Esta abordagem gera uma busca *recursiva* de demonstrações dependentes, mas está implementada no protótipo LEEG.

4.3 Interação entre os Agentes

Esta seção tem por objetivo especificar e detalhar a interação entre os cinco agentes que compõem o sistema LEEG.

A fig.4.10 ilustra o esquema de comunicação entre os cinco agentes que compõem o sistema.

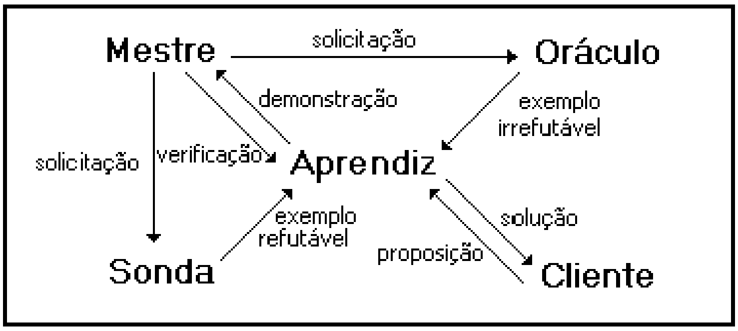

FIGURA 4.10 - Esquema de comunicação do protocolo MOSCA

4.3.1 Interação Cliente ↔ Aprendiz

A comunicação entre os agentes Cliente e Aprendiz estabelece o inicio do processo de aprendizagem. O Cliente envia ao Aprendiz (Cliente → Aprendiz) uma proposição, dentre as proposições disponíveis para demonstração, e aguarda o resultado.

Na proposição escolhida, deve-se identificar a hipótese e a tese. A hipótese deve ser utilizada pelo Aprendiz como enunciado inicial na demonstração. A tese deve ser provada por uma seqüência de enunciados deduzidos logicamente e deverá ser o último enunciado da demonstração.

Uma vez concluída a demonstração, o Aprendiz retorna ao Cliente (Cliente ← Aprendiz) o resultado da prova, apresentando a demonstração completa.

O Cliente tem acesso à base de proposições, que é o conjunto de todas as proposições possíveis que serão submetidas ao Aprendiz. Entretanto, a disponibilização das proposições obedece à relação de dependência entre elas (seção 4.2.3).

4.3.2 Interação Mestre ↔ Aprendiz

A interação entre os agentes Mestre e Aprendiz está relacionada com a construção (Mestre ← Aprendiz) e verificação (Mestre → Aprendiz) da aprendizagem. Cada estado da demonstração desenvolvida pelo Aprendiz é enviada ao Mestre para que este verifique a construção. Três situações podem ocorrer: construção correta, construção do enunciado errada ou construção da regra de dedução errada.

Se a construção estiver correta, o Mestre envia um sinal ao Aprendiz, habilitando o próximo passo da construção.

Se a construção da regra de dedução ou de enunciado estiver errada, o Mestre não habilita o próximo passo da construção ao Aprendiz e envia uma solicitação ao agente Oráculo e Sonda, respectivamente, para que encaminhem mensagem ao Aprendiz. Estas interações estão tratadas nas seções seguintes.

A verificação dos passos desenvolvidos pelo Aprendiz é feita através da comparação entre sua demonstração e a demonstração contida na base de conhecimento do sistema.

4.3.3 Interação Mestre → Oráculo

A comunicação entre os agentes Mestre e Oráculo é feita num único sentido e está relacionada com a solicitação, por parte do Mestre ao Oráculo, de envio de mensagens ao Aprendiz. Esta sinalização ocorre em duas situações: construção errada de regra de dedução e construção correta.

Se Mestre identifica erro na regra de dedução construída pelo Aprendiz, então sinaliza ao Oráculo para que envie ao Aprendiz uma mensagem de correção sobre o referido estado da demonstração.

Por outro lado, se o Mestre verifica correção no estado corrente da demonstração desenvolvida pelo Aprendiz e após o sinal de habilitação nenhuma ação por parte do Aprendiz for identificada pelo Mestre, um sinal é enviado ao Oráculo para que este envie uma mensagem de incentivo que conduza à construção correta.

4.3.4 Interação Mestre → Sonda

Da mesma forma que com o Oráculo, a interação do Mestre com Sonda se dá num único sentido. A comunicação é estabelecida através da emissão de um sinal, do Mestre para o Sonda, solicitando o envio de mensagens de reflexão ao Aprendiz.

Esta sinalização é feita sempre que o Mestre identificar um erro de enunciado na construção do Aprendiz. As mensagens de reflexão têm o objetivo de ajudar o Aprendiz a identificar e corrigir seu erro.

4.3.5 Interação Oráculo → Aprendiz

A comunicação estabelecida entre os agentes Oráculo e Aprendiz está relacionada com os exemplos utilizados para auxiliar a aprendizagem. O Oráculo tem como objetivo enviar ao Aprendiz mensagens que o auxiliem na escolha de regras de dedução corretas, para cada passo da demonstração (mensagens de correção) ou mensagens que incentivem a continuação da demonstração quando identificado abandono da mesma (mensagens de incentivo).

As mensagens enviadas pelo Oráculo ao Aprendiz estão apresentadas na seção 4.2.2.

O Oráculo possui acesso a um Módulo de Exemplos Irrefutáveis (inserido na Base de Exemplos) que é acessado de acordo com o estado corrente da demonstração e mediante sinalização do Mestre.

4.3.6 Interação Sonda → Aprendiz

O agente Sonda tem a função de enviar ao agente Aprendiz mensagens de reflexão, sempre que o Mestre identificar incorreção na construção de enunciados. Estas mensagens de reflexão são determinadas em função da sinalização do Mestre, que identifica o estado corrente da demonstração e o erro cometido pelo Aprendiz.

O agente Sonda tem acesso a um Módulo de Exemplos Refutáveis (também inserido na Base de Exemplos), onde cada mensagem está associada a um estado da demonstração.

4.4 Considerações Finais

Este capítulo descreveu a concepção do sistema LEEG, especificando detalhes referentes à estrutura de dados, classificação de mensagens e interação entre os agentes.

Apresentou-se a modelagem adotada para estruturar os dados do sistema. Para a Base de Conhecimento, foi utilizado o modelo de Autômato Finito com Saída, que permitiu organizar as demonstrações de forma a preservar sua estrutura lógica-dedutiva e facilitando a verificação dos passos elaborados pelo Aprendiz. Mostrou-se ainda a estrutura da Base de Exemplos, com sua subdivisão e com uma classificação das mensagens enviadas pelos agentes Oráculo e Sonda, de acordo com os erros identificados pelo Mestre. Por fim, foi tratada a Base de Proposições, onde se estabeleceu a forma de escolha de proposições estrutura e mostrou-se a dependência das cinco primeiras proposições para ilustrar a abordagem.

A seguir, foram especificadas as relações de comunicação entre os cinco agentes do sistema e a relação destes com as bases as quais possuem acesso. A fig.4.11 mostra essas relações.

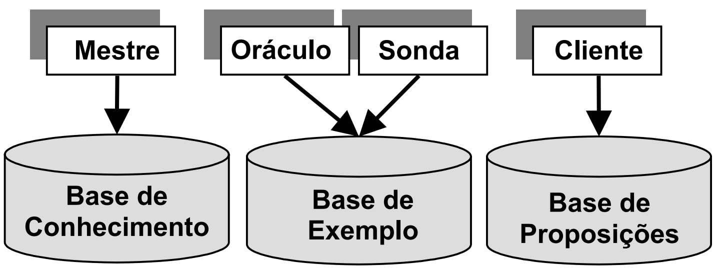

FIGURA 4.11 - Relação agente-base

Este capítulo resultou na publicação de dois artigos enviados para eventos internacionais: *"Hyper-Automaton System Applied to Geometry Demonstration Environment"* e *"Knowledge Anticipation on Agents Relationship in the Geometry Demonstration Proof Environment"* [MAC 01] e [NOT 01a].

No capítulo que segue, serão apresentadas questões relacionadas à implementação física de um protótipo do sistema LEEG, como a linguagem de programação utilizada, estrutura interna dos dados do protótipo, emissão de mensagens e esquema de comunicação entre os agentes

5 Implementação do Protótipo LEEG

Este capítulo apresenta questões referentes à implementação física do sistema LEEG, como a linguagem de programação utilizada, estrutura interna dos dados do sistema, emissão de mensagens e esquema de comunicação entre os agentes.

Esta fase do trabalho foi desenvolvida com a colaboração de um aluno de Iniciação Científica, que contribuiu na implementação do protótipo.

A linguagem de programação utilizada para desenvolver o protótipo foi Delphi 3 [CAN 98] [OSI 97], cuja abordagem é baseada em formulários e orientada a objetos. Escolheu-se a linguagem Delphi por oferecer todos os recursos necessários para o desenvolvimento do protótipo aqui proposto.

Algumas soluções de implementação adotadas não estão totalmente de acordo com a especificação do sistema. Entretanto, o funcionamento do mesmo está totalmente coerente com a especificação realizada no capítulo 4.

Foram implementadas as duas primeiras proposições de Euclides, suficientes para validar a proposta deste trabalho.

5.1 Interface

O sistema desenvolvido é simples e de fácil utilização. Sua interface é formada por poucos componentes, ou seja, apenas o necessário para permitir a construção das demonstrações propostas. A fig.5.1 apresenta a tela de abertura do LEEG.

FIGURA 5.1 - Tela de abertura do LEEG

As proposições que podem ser submetidas à demonstração ficam disponibilizadas em um *menu*, localizado no canto superior esquerdo da tela, como mostra a fig.5.2. Como já mencionado no capítulo 4 (seção 4.2.3), a ordem de dependência entre as proposições é respeitada, ficando disponíveis apenas aquelas proposições que não possuem nenhuma proposição antecedente necessária a sua prova ainda não demonstrada. Na medida em que as demonstrações vão sendo concluídas pelo Aprendiz, novas proposições vão sendo disponibilizadas.

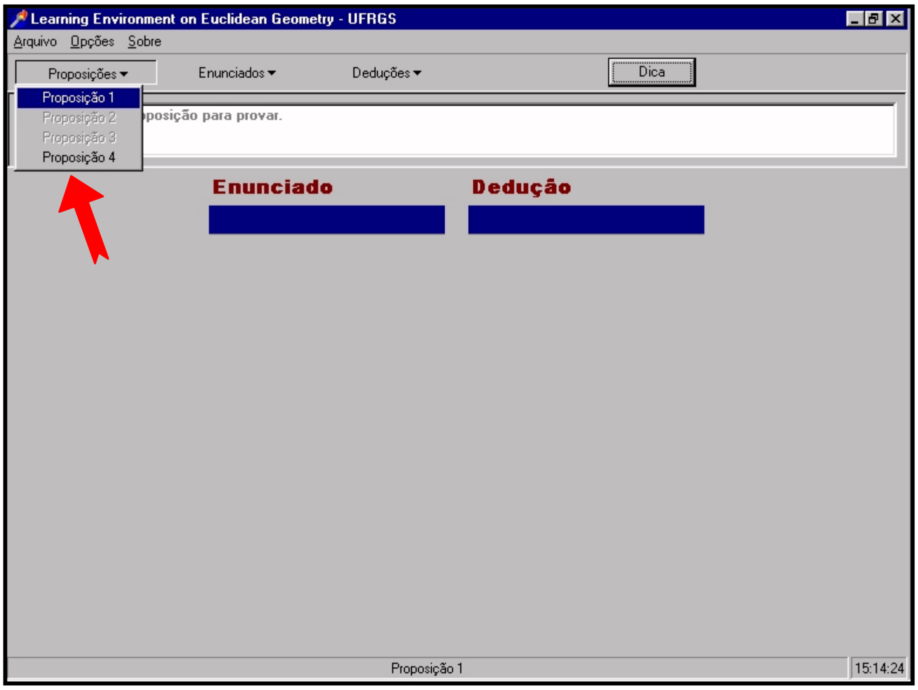

FIGURA 5.2 - Proposições disponíveis para demonstração no LEEG

Os enunciados utilizados para a demonstração das proposições 1 e 2 estão listados em um segundo *menu*, ao lado das proposições. Assim, o Aprendiz pode tanto escrever quanto clicar sobre o enunciado desejado para incluí-lo na demonstração que está construindo. Veja mais detalhes na fig.5.3.

Da mesma forma que os enunciados, as regras de dedução também estão listadas em um *menu*, localizado ao lado do *menu* dos enunciados. Porém, as regras de dedução estão organizadas de forma diferente: num primeiro momento aparecem apenas os grandes grupos de regra de dedução sem especificação, ou seja, axiomas, postulados, definições e proposições. A passar o mouse sobre cada grande grupo, um *submenu* abre-se, apresentando todas as possíveis regras do respectivo grupo (fig.5.4). Quando o mouse passar sobre uma regra de dedução específica, uma explicação é apresentada na barra de *status* (área horizontal localizada logo abaixo da janela principal), que oferece informação sobre o significado ou aplicação da referida regra (fig.5.5).

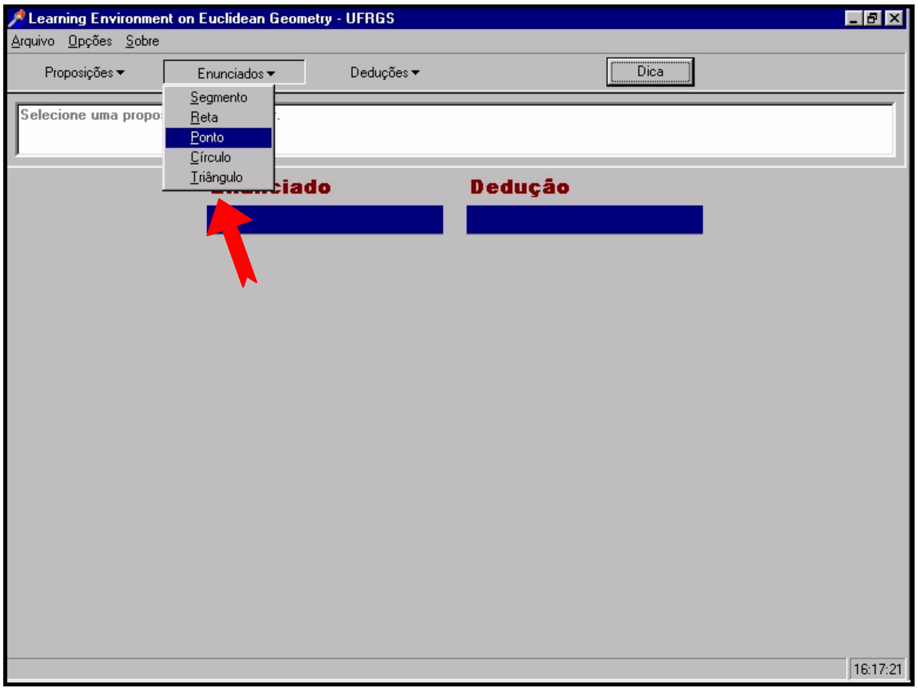

FIGURA 5.3 - Enunciados disponíveis na interface do LEEG

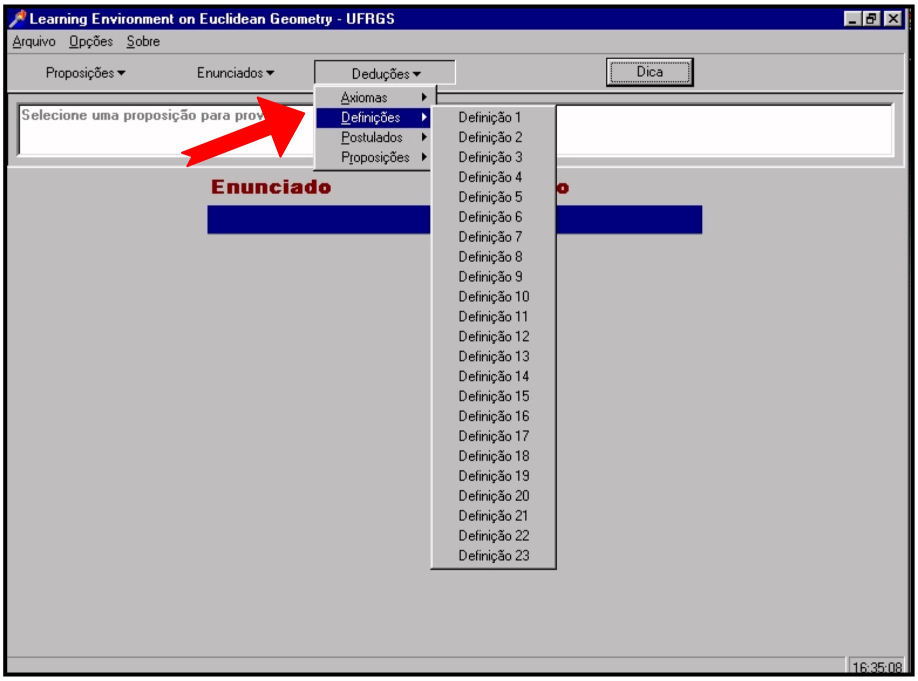

FIGURA 5.4 - Organização das deduções disponíveis na interface do LEEG

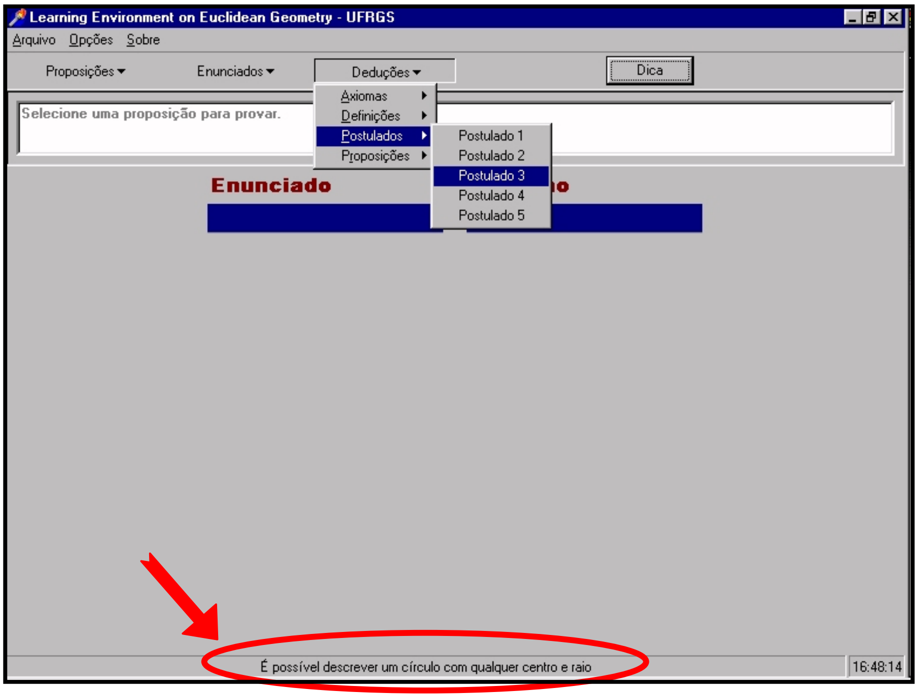

FIGURA 5.5 - Informação na barra de *status*

O processo de demonstração inicia com a seleção de uma proposição. Na barra horizontal superior, apresenta-se o enunciado da proposição selecionada (fig.5.6), de onde se devem extrair a hipótese e tese da mesma. Uma vez selecionada uma das proposições disponíveis para demonstração, os campos da tabela de prova tornam-se habilitados.

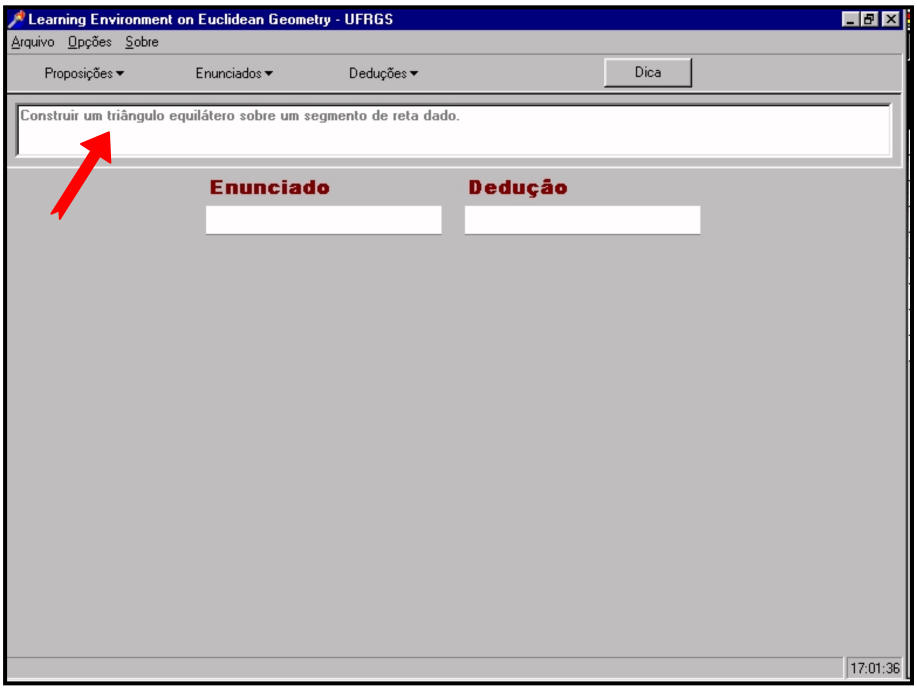

FIGURA 5.6 - Enunciado da proposição selecionada

A demonstração pode iniciar, preenchendo-se os campos da tabela que estão disponíveis. Cada linha da tabela representa um estado da demonstração e só são disponibilizados mediante verificação do estado corrente. A fig.5.7 mostra a demonstração da Proposição 1 sendo construída.

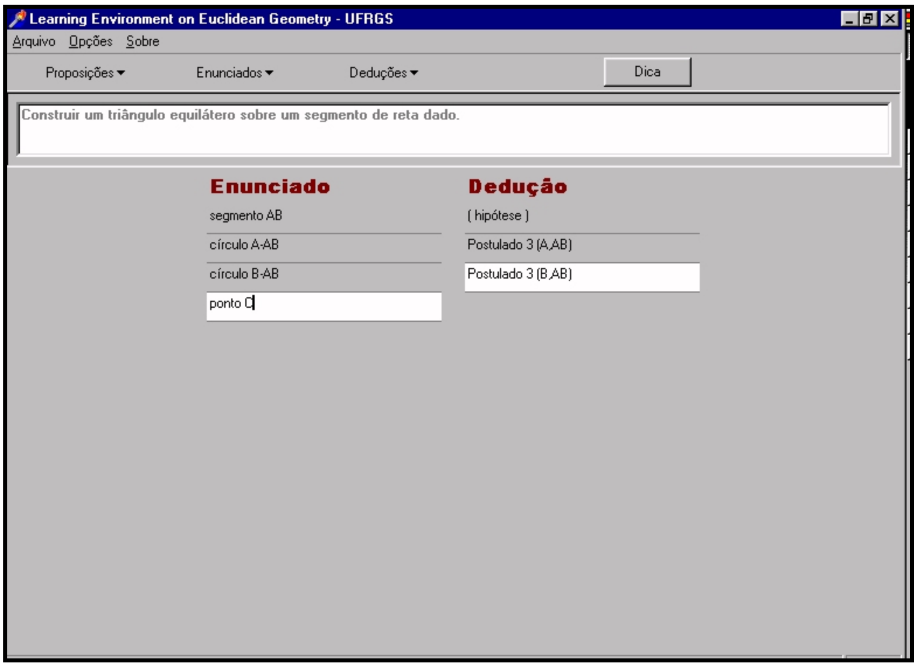

FIGURA 5.7 - Tela do LEEG durante a construção de uma demonstração

Uma vez concluída a demonstração da Proposição 1, a Proposição 2 torna-se disponível, e a Proposição 1 é incluída como uma regra de dedução e pode ser utilizada em todas as demonstrações procedentes que dela forem dependentes.

5.2 Funcionamento Interno

Como já mencionado no capítulo 3, os agentes representados pelo LEEG são o Mestre, Oráculo e Sonda.

O Mestre é responsável pelo acompanhamento da construção da demonstração que está sendo desenvolvida pelo Aprendiz. A cada erro identificado, uma mensagem é enviada para alertar e/ou permitir a correção deste erro.

As mensagens são enviadas pelos agentes Oráculo e Sonda. Porém esses agentes apresentam distinção apenas no nível conceitual do sistema, para permitir a especificação clara de um ambiente de ensino e aprendizagem informatizado com características desejáveis que possibilitem a construção do conhecimento. Entretanto, na implementação da primeira versão do protótipo LEEG, estes agentes não estão representados separadamente.

Seu funcionamento está basicamente focado na leitura e comparação da informação inserida pelo Aprendiz. A fig.5.8 mostra o esquema de funcionamento interno do LEEG.

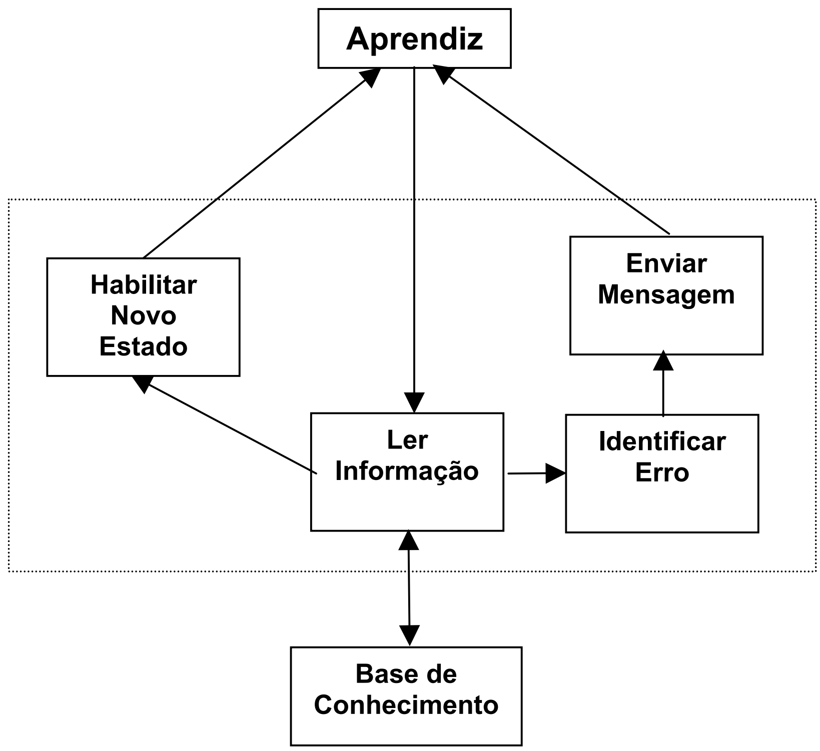

FIGURA 5.8 - Esquema de funcionamento interno

O processo de análise dos enunciados e regras de dedução inseridos pelo Aprendiz na demonstração está ilustrado a seguir (fig.5.9 e fig.5.10, respectivamente), através de um fluxograma simplificado.

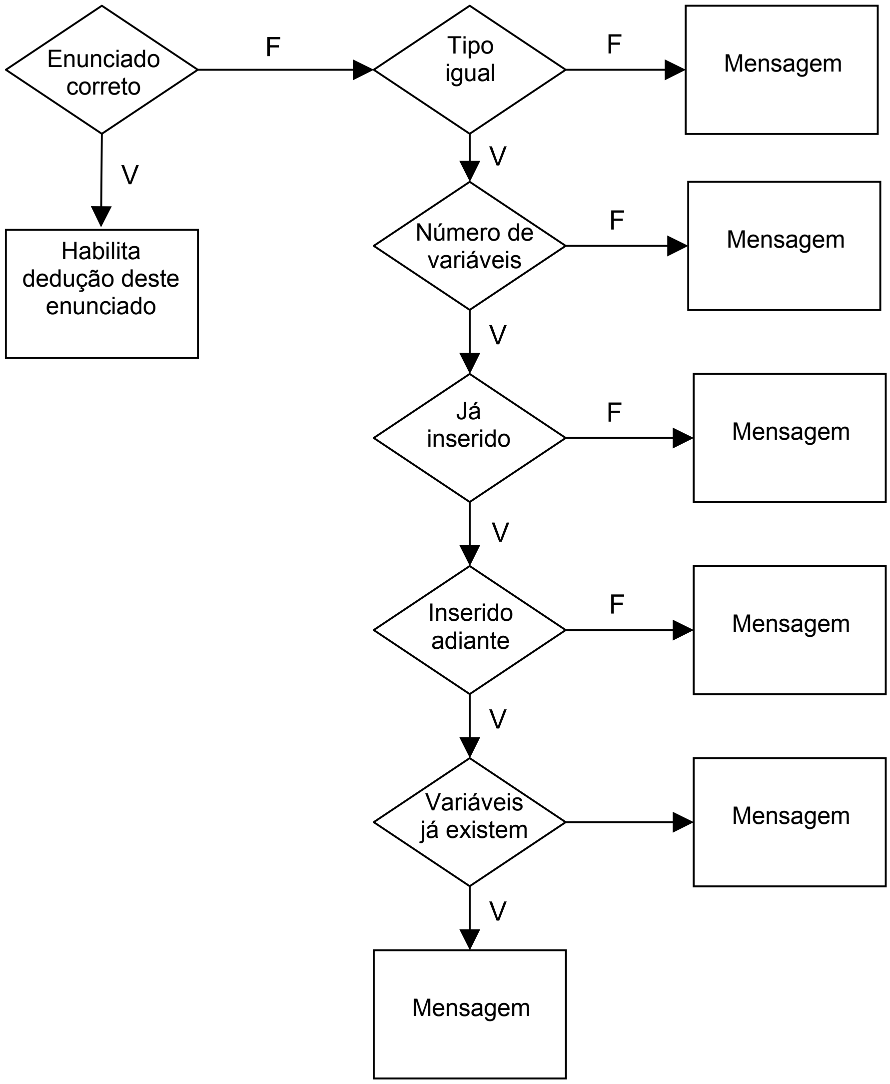

FIGURA 5.9 - Fluxograma de análise dos enunciados

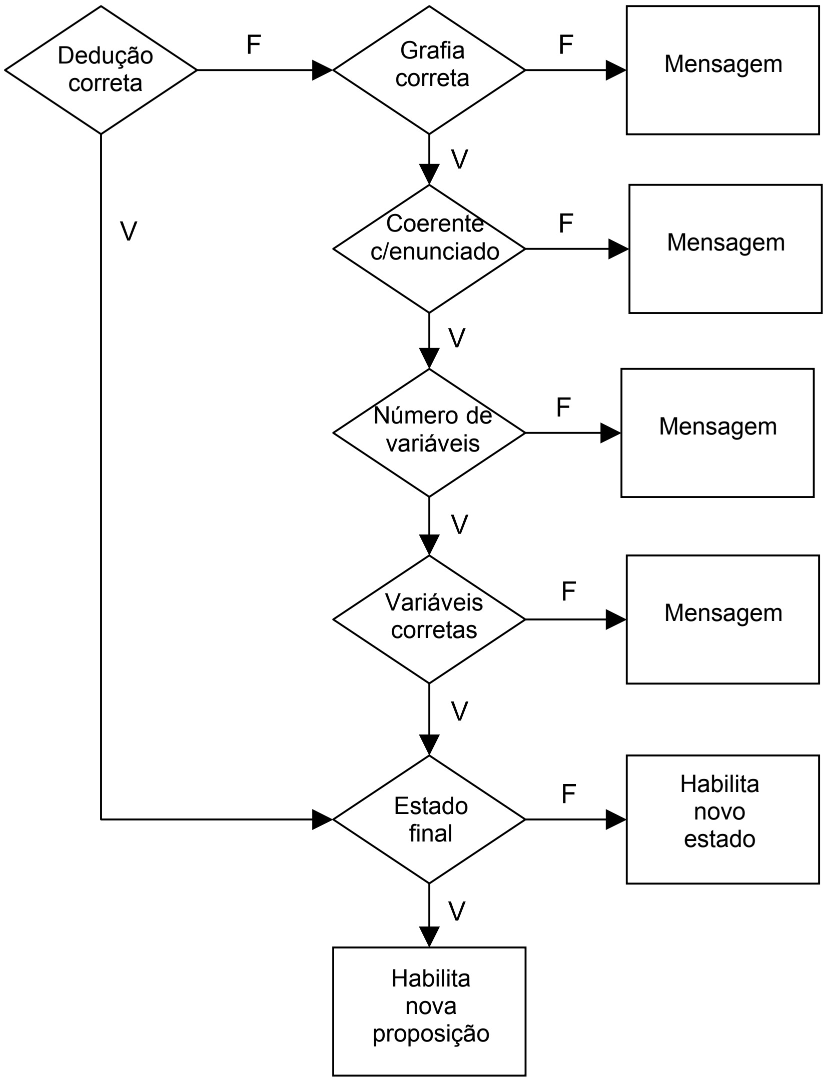

FIGURA 5.10 - Fluxograma de análise das regras de dedução

Quanto às mensagens de incentivo, optou-se por tornar sua utilização opcional, ou seja, o usuário define se serão ou não enviadas as mensagens que contém dicas relativas ao próximo estado, passado os 10 minutos com ausência de ação do Aprendiz (definido no capítulo 4). É possível ainda habilitar ou não o botão "dica" que, quando acionado pelo Aprendiz, encaminha a mensagem de incentivo relativa ao estado corrente.

Sempre que o Aprendiz sai do sistema, toda sua construção é gravada em um arquivo, para que, na próxima utilização, prossiga com as demonstrações exatamente de onde parou pela última vez.

5.3 Base de Conhecimento

A base de conhecimento do sistema, como mencionado no capítulo 4, foi implementada através do modelo de Autômatos Finitos com Saída. Cada demonstração incluída na base de conhecimento é representada por um autômato diferente, que guarda a estrutura dedutiva e todos os possíveis raciocínios que levam da hipótese à tese da proposição.

Cada autômato está gravado em um arquivo do tipo "documento de texto" (.txt), que contém informações sobre cada estado da demonstração, o estado de destino, a função associada a cada transição e a saída associada a cada estado. Estas saídas representam as mensagens de incentivo enviadas pelo Oráculo ao Aprendiz que, para as proposições 1 e 2, estão apresentadas nas tab.4.6 e tab.4.7, respectivamente. A fig.5.11 mostra a especificação de um estado da demonstração da Proposição 1.

```
!3 4
#incentivo
O terceiro vértice do triângulo vem da interseção dos dois círculos.
#enunciado com novos pontos
ponto &2
#deducao
interseção ( &0 - &0 &1 , &1 - &0 &1 )
```

FIGURA 5.11 - Especificação de um estado da demonstração da Proposição 1

Teve-se a preocupação de permitir liberdade na escolha da nomenclatura dos elementos geométricos por parte do usuário. Isto significa que é possível demonstrar uma proposição utilizando qualquer nomenclatura para nomear os elementos utilizados na prova, ou seja, quando se quer incluir um segmento como enunciado da demonstração, pode-se chamá-lo de "segmento AB" ou "segmento XY" ou qualquer outra combinação de dois caracteres de preferência do Aprendiz, que o sistema irá aceitá-lo. Isto porque as variáveis que vão sendo criadas pelo Aprendiz são armazenas em um vetor, na ordem em que são construídas, para respeitar a lógica da dedução e permitir a verificação. Pode-se conferir isso na fig.5.12, que mostra a construção de uma demonstração em duas situações, que adotam variáveis diferentes.

Outra preocupação também considerada na implementação da base foi prever as diferentes formas de representar um mesmo elemento, enunciado ou regra de dedução. Por exemplo, sabe-se que falar em "segmento AB" ou "segmento BA", está-se falando do mesmo segmento, ou afirmar que "segmento CD = segmento EF" é o mesmo que afirmar que "segmento EF = segmento CD" (veja fig.5.13). Dessa forma, todas as possíveis formas de representação são aceitas pelo LEEG, com o objetivo de permitir total liberdade de expressão ao usuário do sistema (Aprendiz).

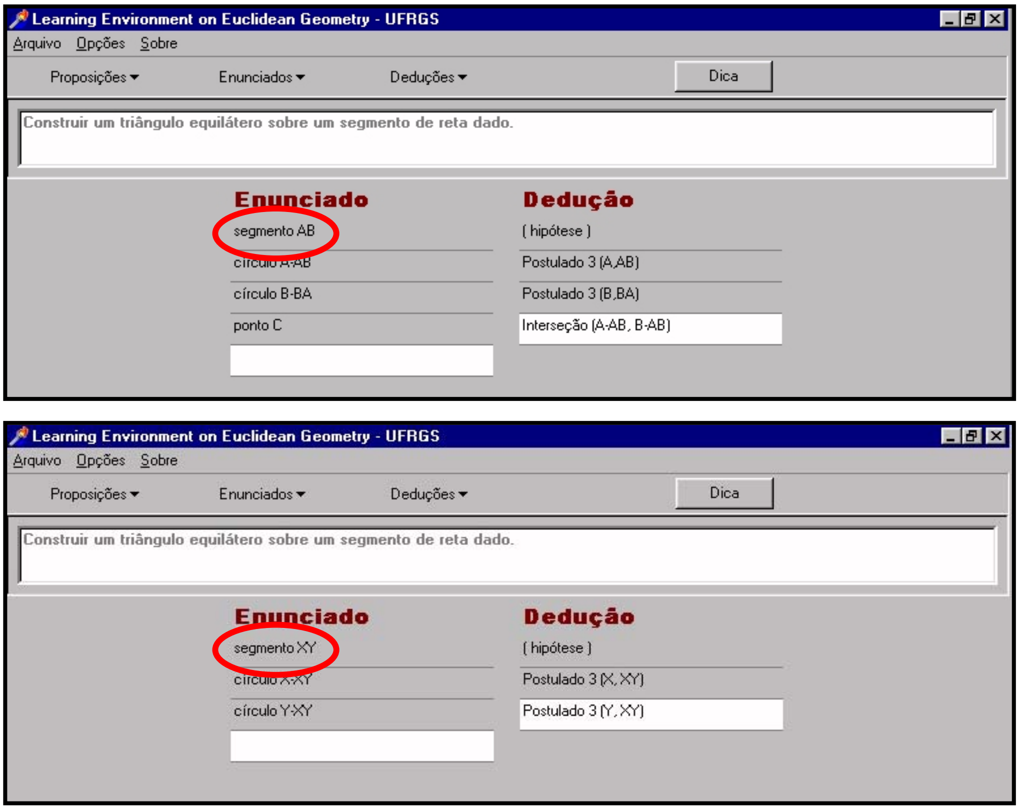

FIGURA 5.12 - Utilização de diferentes nomenclaturas

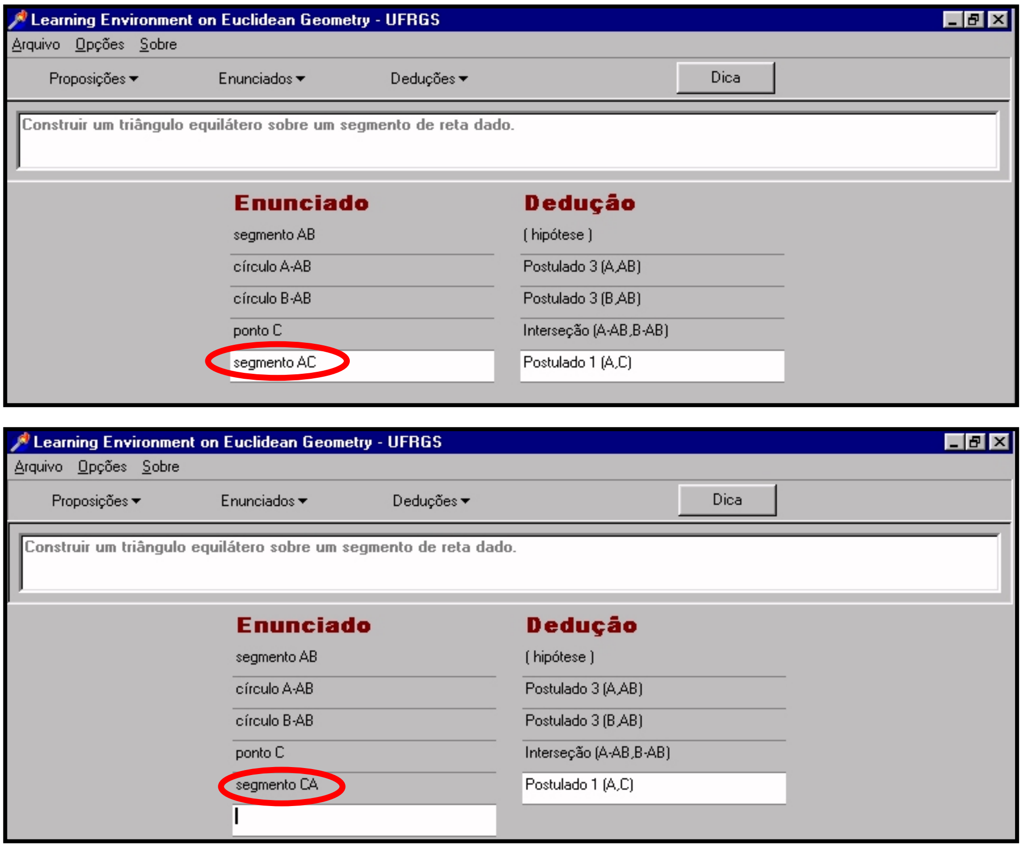

FIGURA 5.13 - Diferentes representações para um mesmo elemento

5.4 Construção de Demonstrações – Um Exemplo

Apresenta-se a seguir um exemplo de construção de demonstração da Proposição 1, que ilustra diversos erros identificados pelo Mestre e as respectivas mensagens enviadas pelo sistema.

O primeiro passo é selecionar a proposição que se pretender demonstrar (no caso a Proposição 1) e, a partir de seu enunciado, identificar a hipótese para iniciar a construção da prova. Um primeiro erro que pode ser cometido é relativo ao número de variáveis que definem determinado elemento. Veja, na fig.5.14 a ilustração desse fato.

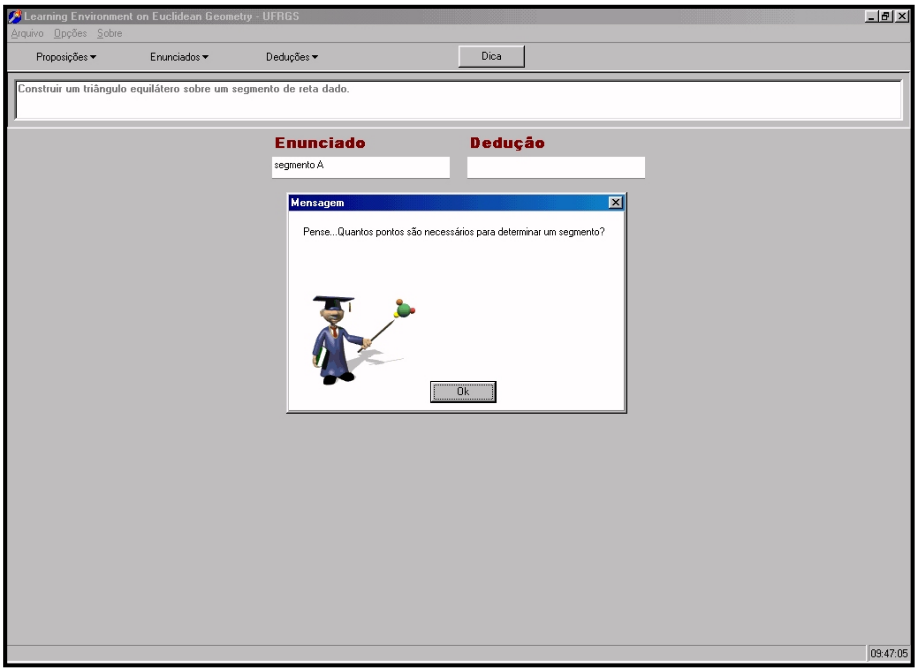

FIGURA 5.14 – Erro relacionado ao número de variáveis

Após a emissão da mensagem de correção, o erro pode ser percebido pelo Aprendiz e corrigido. Na seqüência, pode ser cometido um erro relativo à regra de dedução que deduz o respectivo enunciado, e uma nova mensagem é enviada (fig.5.15).

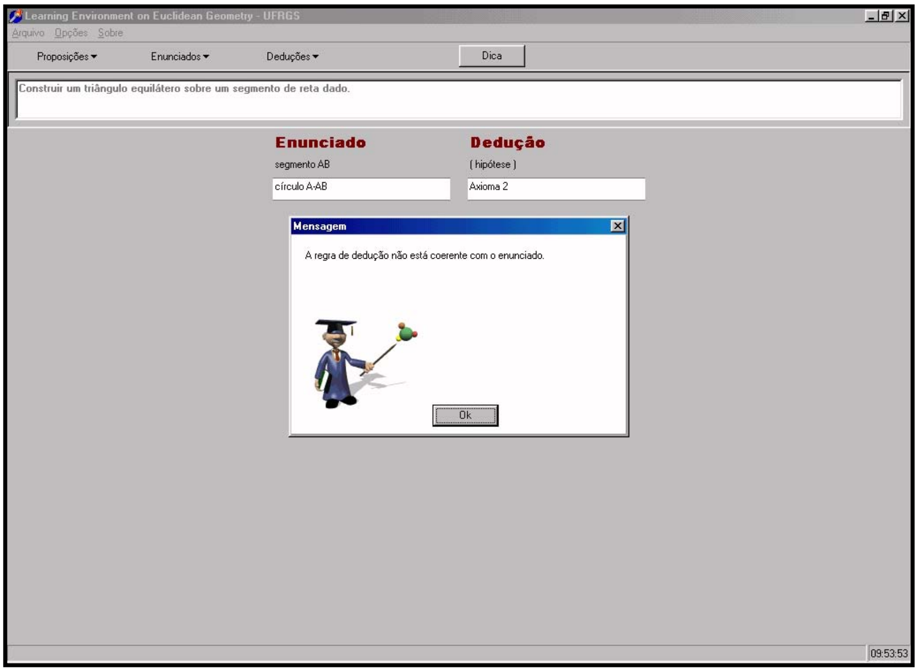

FIGURA 5.15 – Erro relacionado à regra de dedução

Prosseguindo com a construção, pode-se inserir um enunciado que já foi construído anteriormente, e um novo alerta é enviado (fig.5.16).

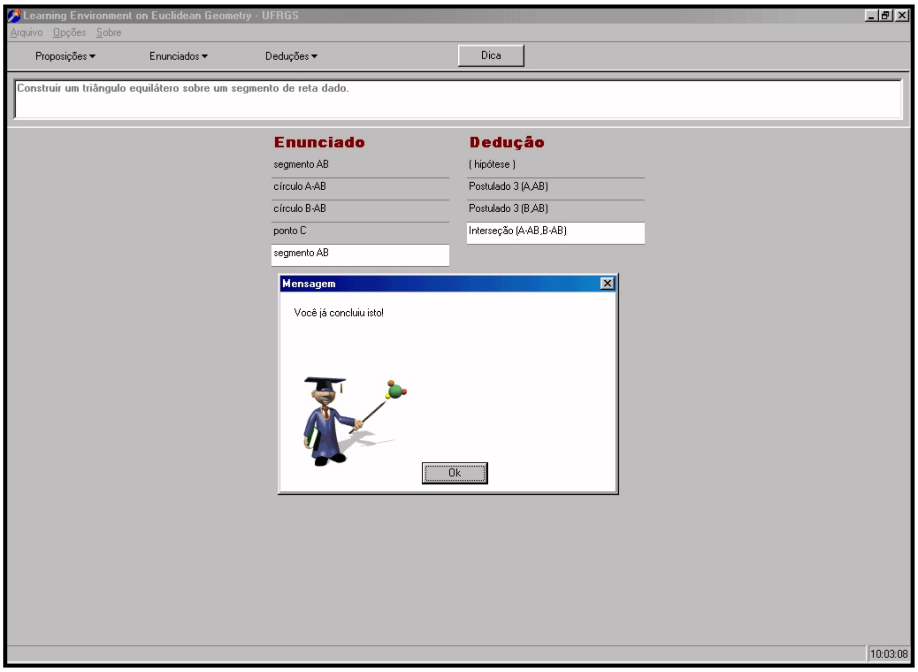

FIGURA 5.16 – Erro de enunciado já inserido

Outro erro que pode ocorrer é quanto à aplicação da regra de dedução às variáveis corretas. Veja na fig.5.17.

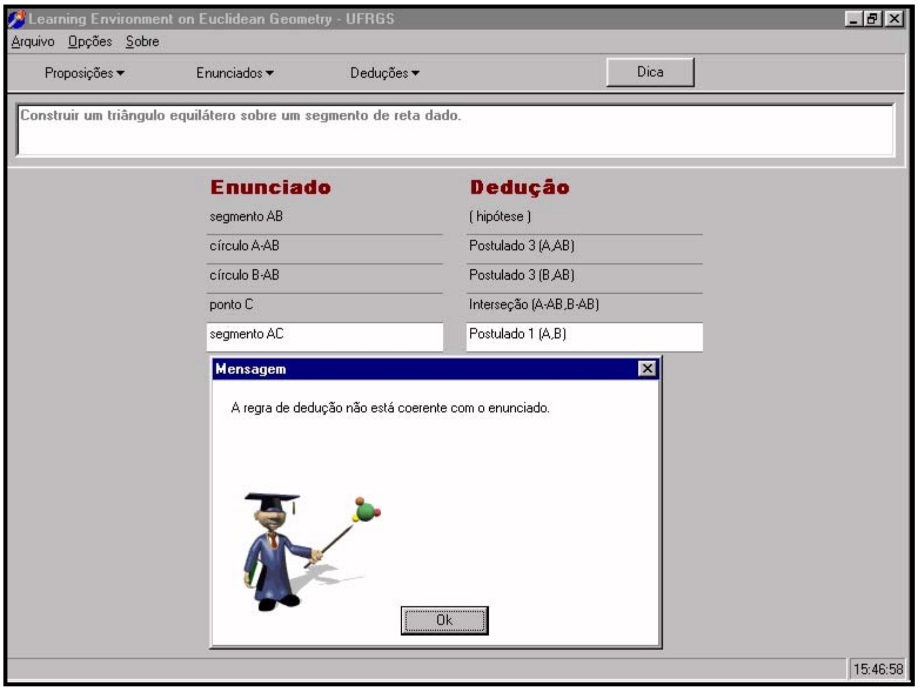

FIGURA 5.17 - Erro variáveis

A figura a seguir mostra uma mensagem alertando quanto a um enunciado inserido precipitadamente (fig.5.18).

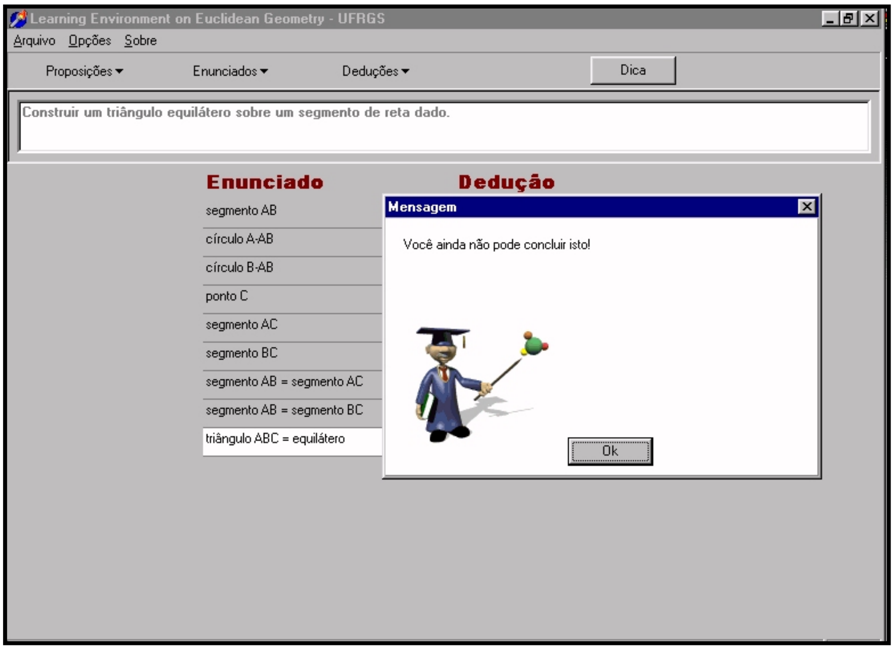

FIGURA 5.18 - Erro de enunciado inserido precipitadamente

Assim que a demonstração é concluída, uma mensagem final é apresentada (fig.5.19).

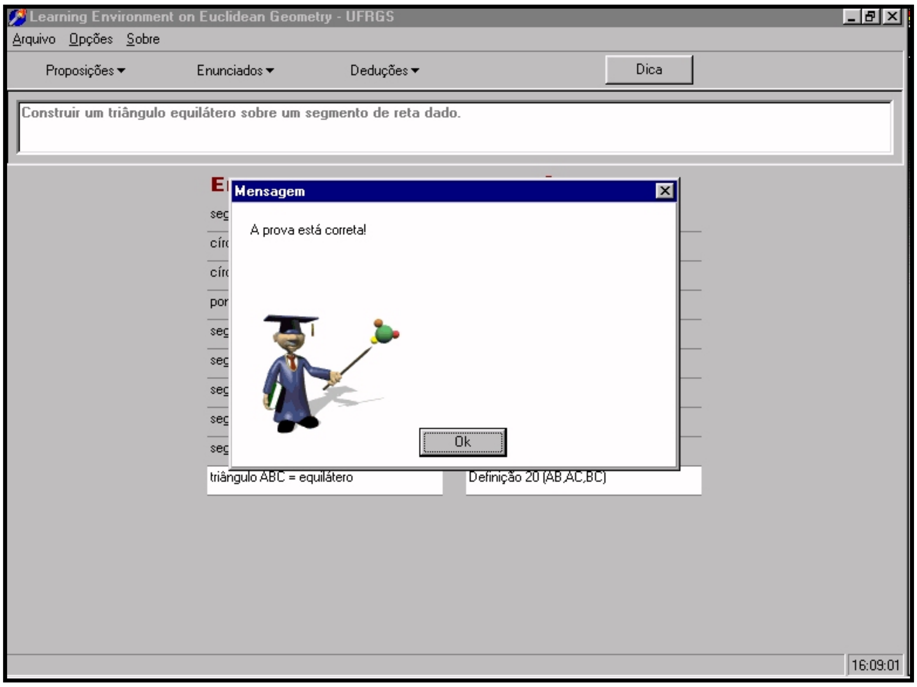

FIGURA 5.19 - Mensagem final

5.5 Considerações Finais

Este capítulo apresentou um breve comentário referente à implementação do protótipo do LEEG, onde descreve sua interface, apresenta considerações relacionadas ao seu funcionamento interno e a sua base de conhecimento.

Por fim, apresentou um exemplo de construção de uma demonstração para a Proposição 1, ilustrando algumas interações realizadas durante o processo.

Como resultado deste capítulo, teve-se a elaboração do artigo *"Sistema de Aprendizagem de Demonstrações da Geometria Euclidiana Plana – LEEG"*, submetido para avaliação e que aguarda parecer até o presente momento.

O próximo capítulo apresenta as conclusões dessa dissertação, trazendo uma breve revisão dos temas tratados nesse volume, os resultados obtidos com o trabalho e trabalhos futuros que darão continuidade e complementarão o que foi até então desenvolvido.

6 Conclusões

Nos últimos anos, uma acelerada evolução dos sistemas de ensino e aprendizagem informatizados tem sido observada. A preocupação com o desenvolvimento de ambientes cada vez mais eficientes, tanto do ponto de vista computacional quanto pedagógico, tem repercutido num salto de qualidade dos software educacionais.

Vem-se utilizando cada vez mais técnicas de Inteligência Artificial para desenvolver sistemas de ensino mediados por computador que se aproximem ao máximo do professor humano, tendo capacidade de adaptar-se às necessidades individuais de cada aluno, levando em consideração seus conhecimentos prévios e suas limitações, ou seja, o perfil de cada aluno.

Por outro lado, sabe-se que a escola precisa acompanhar esta nova era da informação que estamos vivenciando. Proporcionar aos alunos do século XXI, acostumados com jogos computacionais e Internet, sistemas de aprendizagem que ofereçam tecnologia com qualidade de ensino é para onde caminha a Educação.

Indo mais além, permitir aos alunos construir seu conhecimento mesmo na ausência do professor é um passo importante que se está dando.

No contexto do ensino de Matemática, a aprendizagem de demonstrações da Geometria Euclidiana apresenta um histórico de dificuldades e, conseqüentemente, abandono. Grande parte dos professores não desenvolve demonstrações geométricas com seus alunos. A dificuldade apresentada na aprendizagem de demonstrações é um fato constatado, uma vez que exige um grau de abstração não atingido por um significativo número de estudantes.

No que diz respeito aos software desenvolvidos para o ensino e aprendizagem de demonstrações em Geometria Euclidiana, poucos são encontrados. Na verdade, o que se encontra são programas que realizam demonstrações geométricas através de métodos algébricos, mas que não apresentam a mínima preocupação com o processo de construção do conhecimento, ou seja, com o processo de ensino-aprendizagem.

Sabe-se que a aprendizagem de demonstrações em Geometria pode ser subsidiada por dicas que instiguem o raciocínio do aprendiz e encaminhem à dedução correta. Oferecer aos estudantes um ambiente para experimentação das idéias ou para construção delas pode ser um passo para trazer novamente o ensino das provas geométricas.

Desta forma, esta dissertação de mestrado teve como objetivo desenvolver um sistema para a aprendizagem de demonstrações da Geometria Euclidiana Plana, baseada na aprendizagem através de exemplos.

Desenvolveu-se o sistema LEEG (*Learning Environment on Euclidean Geometry*), que se caracteriza por auxiliar a aprendizagem das provas geométricas desenvolvidas por Euclides, na sua famosa obra *Elements*. Optou-se pelas demonstrações feitas por Euclides, por serem demonstrações totalmente desenvolvidas através do pensamento geométrico, que utilizam elementos básicos da geometria (ponto, reta, círculo,...) e construção elementares com régua e compasso para desenvolver, de forma surpreendente, toda a Geometria que é conhecida e estudada até os dias de hoje.

Como já mencionado anteriormente, os software até então desenvolvidos, constroem provas que apelam para a total manipulação algébrica. Apesar da extrema importância destes software, seus objetivos são outros: o desenvolvimento de sistemas capazes de produzir provas de teoremas da Geometria, mas que não tem a preocupação com o ensino e aprendizagem.

A proposta deste trabalho foi desenvolver um sistema que ofereça condições para a construção e aprendizagem da demonstração desses teoremas por aprendizes humanos.

Para o desenvolvimento deste trabalho, foram estudados temas considerados requisitos indispensáveis para sua elaboração, como: Geometria Euclidiana, demonstração de teoremas, teoria de agentes e autômatos finitos com saída.

A concepção do sistema foi realizada sobre uma adaptação do protocolo de aprendizagem MOSCA, que define um ambiente de aprendizagem baseado na utilização de exemplos e que é formado por cinco agentes: Mestre, conhecedor do conhecimento que se está ensinando e responsável pela permanente verificação dos passos construídos pelo agente Aprendiz (um dos cinco agentes que compõem o ambiente e que é, como o próprio nome diz, aquele que deve aprender as demonstrações), Oráculo e Sonda, responsáveis pela emissão dos exemplos e Cliente, que submete as proposições ao Aprendiz para serem demonstradas.

A base de conhecimento do sistema, que guarda a estrutura lógica-dedutiva de todas as demonstrações que podem ser submetidas ao Aprendiz, foi implementada através do modelo de autômatos finitos com saída. A utilização de autômatos com saída na aplicação de modelagem de demonstrações dedutivas foi extremamente útil por permitir estruturar os diferentes raciocínios que levam da hipótese à tese da proposição de forma lógica, organizada e direta. Outra importante vantagem na utilização de autômatos para modelar a base do sistema é a facilidade oferecida na verificação da construção que está sendo desenvolvida pelo Aprendiz, já que cada passo da demonstração é representado por um estado do autômato e cada regra de dedução e representada pela função de transição no autômato. Dessa forma, sabe-se o estado exato em que o Aprendiz se encontra e pode-se associar dicas para cada um desses estados. Ainda tem-se a possibilidade guardar o estado onde o Aprendiz parou sua demonstração para, posteriormente, prosseguir a prova do estado em que parou.

Quanto à produção científica, o trabalho desenvolvido resultou na produção e publicação de artigos nacionais e internacionais, tanto em nível de autoria quanto co-autoria, em eventos relacionados aos temas que norteiam o trabalho. Até o momento, teve-se: três artigos publicados em caráter regimental ou regional; cinco artigos publicados em congressos, na versão completa ou resumida, dois artigos que serão publicados (em 2002) em capítulos de livro e um artigo que será publicado em periódico (também em 2002). São eles:

- **Uso do Computador na Educação: Um Histórico Brasileiro**. *Simpósio de Informática do Planalto Médio*, 2., 2000, Passo Fundo. [NOT 00]
- **Evolução Cronológica da Informática Educativa na Informática da UFRGS**. *Trabalho Individual I*, 2000. [NOT 00a]
- **Considerações sobre o Raciocínio de um Agente Racional**. *Semana Acadêmica do PPGC*, 2000. [NOT 00b]
- **Hyper-Automaton System Applied to Geometry Demonstration Environment**. *International Conference on Computer Aided Systems Theory and Technology*, 8., 2001, Las Palmas de Gran Canaria. [MAC 01]
- **Hyper-Automaton System Applied to Geometry Demonstration Environment**. *Lecture Notes in Computer Science*, v. 2178. Versão estendida do artigo. [MAC 01a]
- **Aprendiz's Learning in Geometry Demonstration**. *World Conference on Computer in Education*, 7., 2001, Copenhagen. [NOT 01]
- **Historical Overview of Computer usage in Brazilian Teaching**. *World Conference on Computer in Education*, 7., 2001, Copenhagen. [NOT 01b]
- **Historical Overview of Computer usage in Brazilian Teaching**. *Pos Conference Book WCCE 2001*. Versão estendida do artigo. [NOT 01c]
- **Knowledge Anticipation on Agents Relationship in the Geometry Demonstration Proof Environment**. *International Conference on Computing Anticipatory Systems*, 5., 2001, Liège. [NOT 01a]
- **Knowledge Anticipation on Agents Relationship in the Geometry Demonstration Proof Environment**. *International Journal of Computing Anticipatory Systems.* Versão estendida do artigo. [NOT 01d]
- **Sistema de Aprendizagem de Demonstrações da Geometria Euclidiana Plana - LEEG**. *Simpósio Brasileiro de Informática na Educação*, 12., 2001, Vitória.

Ainda como resultado da dissertação de mestrado, tem-se a primeira versão do sistema LEEG. Entretanto, essa versão apresenta algumas limitações, que devem ser resolvidas em trabalhos futuros.

Apenas duas proposições foram implementadas para validar a proposta da dissertação. Assim, o próximo passo, é implementar as demais proposições da Geometria Euclidiana Plana.

Considerar o agente Aprendiz artificial e analisar seu raciocínio durante a aprendizagem das demonstrações também é uma tarefa que está sendo proposta como trabalho futuro.

Quanto às características do sistema, um questionário de avaliação será elaborado para ser submetido a um grupo de pessoas que deverão utilizar o sistema e analisá-lo. Esta avaliação subsidiará as modificações futuras as quais será submetida a primeira versão do protótipo do sistema.

Aprofundar os conhecimentos no que diz respeito às teorias da aprendizagem e à psicologia do desenvolvimento também faz parte dos projetos futuros, visando melhorar a qualidade pedagógica do software, assim como assimilar técnicas de IA para tornar o sistema adaptativo ao perfil de cada estudante.

ANEXO 1 Proposições da Geometria Euclidiana Plana

Proposição 1: Construir um triângulo equilátero sobre um segmento de reta dado.

Proposição 2: Construir um segmento de reta igual a um segmento de reta dado com uma das extremidades em ponto dado.

Proposição 3: Dados dois segmentos desiguais, cortar, do segmento maior, um segmento igual ao segmento menor.

Proposição 4: Se dois triângulos têm dois lados iguais respectivamente, e têm os ângulos formados por esses lados também iguais, então eles também têm as bases iguais, os triângulos são iguais e os ângulos restantes são iguais respectivamente, chamados de ângulos opostos a lados iguais.

Proposição 5: Em um triângulo isósceles, os ângulos da base são iguais e, se as linhas retas iguais são prolongadas, então os ângulos abaixo da base são iguais.

Proposição 6: Se em um triângulo dois ângulos são iguais entre si, então os lados opostos aos ângulos iguais também são iguais um ao outro.

Proposição 7: Dados dois segmentos construídos sobre as extremidades de um segmento e que se encontram em um ponto, é impossível construir sobre as extremidades do mesmo segmento e sobre o mesmo lado dele, dois outros segmentos que se encontram em outro ponto e que sejam iguais aos dois segmentos dados.

Proposição 8: Se dois triângulos têm dois lados respectivos iguais e se têm bases iguais entre si, então eles possuem os ângulos formados pelos lados iguais também iguais.

Proposição 9: Dividir um ângulo dado em dois ângulos iguais (bissetriz).

Proposição 10: Dividir um segmento dado em dois segmentos iguais (mediatriz).

Proposição 11: Construir uma reta perpendicular a um segmento dado passando por um por dado sobre este segmento.

Proposição 12: Construir uma reta perpendicular a uma reta dada passando por um ponto dado fora da reta.

Proposição 13: Se um segmento corta outro segmento, então ele forma dois ângulos retos ou ângulos cuja soma é igual a dois ângulos retos (180º).

Proposição 14: Se a partir de qualquer segmento e um ponto sobre ele, dois segmentos que não se situam sobre o mesmo lado dele formam dois ângulos adjacentes cuja soma é igual a dois ângulos retos (180º), então os dois segmentos estão alinhados um ao outro (formam uma única reta).

Proposição 15: Se duas retas se interceptam, então os ângulos opostos são iguais.

Corolário: Se duas retas se interceptam, então os ângulos formados nos pontos de interseção são iguais a quatro ângulos retos (360º).

Proposição 16: Em qualquer triângulo, se um dos lados é prolongado, então o ângulo externo é maior que os ângulos internos opostos.

Proposição 17: Em qualquer triângulo a soma de quaisquer dois ângulos é menor que dois ângulos retos (180º).

Proposição 18: Em qualquer triângulo, o ângulo oposto ao maior lado é o maior ângulo.

Proposição 19: Em qualquer triângulo, o lado oposto ao maior ângulo é o maior lado.

Proposição 20: Em qualquer triângulo, a soma de quaisquer dois lados é maior que o terceiro lado.

Proposição 21: Se a partir dos vértices de um dos lados de um triângulo dois segmentos são construídos, encontram-se em um ponto interno ao triângulo, então a soma dos segmentos construídos é menor que a soma dos dois lados restantes do triângulo, mas os segmentos construídos formam um ângulo maior que o ângulo formado pelos dois lados restantes.

Proposição 22: Construir um triângulo com lados iguais a três segmentos dados: mas é necessário que a soma de qualquer dois dos lados seja maior que o terceiro lado.

Proposição 23: Construir um ângulo igual a um ângulo dado sobre uma reta dada e um ponto sobre ela.

Proposição 24: Se dois triângulos têm dois lados respectivos iguais, mas contêm um dos ângulos formados pelos lados iguais maior que o outro, então eles têm uma base maior que a outra base.

Proposição 25: Se dois triângulos têm dois lados respectivos iguais, mas têm uma base maior que a outra, então eles têm um dos ângulos formados pelos lados iguais maior que o outro.

Proposição 26: Se dois triângulos têm dois ângulos respectivos iguais e um dos lados igual (o lado contido entre os dois ângulos iguais), então os dois lados restantes são iguais respectivamente, e o ângulo restante igual.

Proposição 27: Se uma reta corta outras duas retas formando ângulos alternos iguais entre si, então as duas retas são paralelas.

Proposição 28: Se uma reta corta outras duas retas formando o ângulo externo igual ao ângulo interno oposto, ou a soma dos ângulos internos do mesmo lado é igual a dois ângulos retos (180º), então as duas retas são paralelas.

Proposição 29: Uma reta corta retas paralelas e forma ângulos alternos iguais, o ângulo externo é igual ao ângulo interno oposto, e a soma dos ângulos internos do mesmo lado é igual a dois ângulos retos (180º).

Proposição 30: Retas paralelas a uma mesma reta são paralelas entre si.

Proposição 31: Construir uma reta, passando por um ponto dado, paralela a uma reta dada.

Proposição 32: Em qualquer triângulo, se um dos lados é prolongado, então o ângulo externo é igual à soma dos dois ângulos internos opostos, e a soma dos três ângulos internos do triângulo é igual a dois ângulos retos (180º).

Proposição 33: Segmentos que unem extremidades de segmentos iguais e paralelos na mesma direção também são iguais e paralelos.

Proposição 34: Em um paralelogramo, os lados opostos e os ângulos opostos são iguais, e o diâmetro divide a área em partes iguais.

Proposição 35: Paralelogramos sobre uma mesma base e sobre as mesmas paralelas são iguais.

Proposição 36: Paralelogramos com bases iguais e sobre as mesmas paralelas são iguais.

Proposição 37: Triângulos sobre uma mesma base e compreendidos entre retas paralelas são iguais.

Proposição 38: Triângulos com bases iguais e compreendidos entre paralelas são iguais.

Proposição 39: Triângulos iguais que estão sobre a mesma base e compreendido entre o mesmo lado, estão entre as mesmas paralelas.

Proposição 40: Triângulos iguais com bases iguais e compreendido entre o mesmo lado, estão entre as mesma paralelas.

Proposição 41: Se um paralelogramo tem a mesma base que um triângulo e se ambos estão compreendidos entre as mesmas paralelas, então o paralelogramo é o dobro do triângulo.

Proposição 42: Construir um paralelogramo igual a um triângulo dado sobre um ângulo dado.

Proposição 43: Em qualquer paralelogramo, os complementos (paralelogramos com lados paralelos ao paralelogramo original, com um dos vértices sobre o diâmetro) formados pelo diâmetro são iguais.

Proposição 44: Sobre um segmento dado e um ângulo dado, construir um paralelogramo igual a um triângulo dado.

Proposição 45: Construir um paralelogramo igual a uma figura dada sobre um ângulo dado.

Proposição 46: Descrever um quadrado sobre um segmento dado.

Proposição 47: Em triângulos retângulos, o quadrado construído sobre o lado oposto ao ângulo reto é igual à soma dos quadrados construídos sobre os lados que contém o ângulo reto.

Proposição 48: Se em um triângulo o quadrado construído sobre um dos lados é igual à soma dos quadrados construídos sobre os dois lados restantes, então o ângulo contido pelos dois lados restantes é reto.

ANEXO 2 Demonstração das cinco Primeiras Proposições

Estão apresentadas aqui as cinco primeiras demonstrações da Geometria Euclidiana Plana, com comentários e suas relações de dependência.

A Proposição 1 independe de outras proposições para ser demonstrada. Apenas os Postulados 1 e 3, as Definições 15 e 20 e o Axioma 1 são utilizados para deduzir sua tese. A Proposição 2, entretanto, depende da Proposição 1, além de depender dos Postulados 1, 2 e 3, das Definições 15 e 20 e dos Axiomas 1 e 3. Já a Proposição 3 depende da Proposição 2, do Postulado 3, da Definição 15 e do Axioma 1. A Proposição 4 depende apenas do Axioma 4. Finalmente a Proposição 5, que depende das Proposições 3 e 4 para ser demonstrada, além de depender da Definição 20, dos Postulados 1 e 2 e do Axioma 3. Para uma melhor visualização das relações de dependência entre os termos considerados, veja a tab.1 a seguir.

TABELA 1 - Relações de dependência dos termos utilizados para a prova da Proposição 5

	Definição	Postulado	Axioma	Proposição
Proposição 1	15 - 20	1 - 3	1	
Proposição 2	15 - 20	1 - 2 - 3	1 - 3	1
Proposição 3	15	3	1	2
Proposição 4			4	
Proposição 5	20	1 - 2	3	3 - 4

A seguir, tem-se a demonstração das proposições de 1 a 5.

Proposição 1: *Construir um triângulo equilátero sobre um segmento de reta dado.*

Hipótese: segmento AB

Tese: Triângulo ABC equilátero

Demonstração: Seja AB o segmento dado. Descreva um círculo C1 com centro A e raio AB e um círculo C2 com centro B e raio BA (o *Postulado 3* garante a construção de um círculo a partir de um centro e um raio). A interseção dos círculos determina o ponto C. Em seguida, traçar os seguimentos AC e BC (o *Postulado 1* garante que dois pontos determinam um segmento). Como o ponto A é o centro do círculo C1, e os pontos B e C pertencem a este círculo, então AC é igual a AB (a *Definição 15* diz que se dois segmentos são raios de um mesmo círculo, então estes segmentos são iguais). Da mesma forma, como o ponto B é centro do círculo C2, e os pontos A e C pertencem a ele, então BC é igual a BA (*Definição 15*). Mas se AC é igual a AB e BC é igual a AB (já provado), então AC e BC são iguais a AB. Mas então AC é igual a BC (pelo *Axioma 1*, que diz que objetos que são iguais a um mesmo objeto, são iguais entre si). Logo, os três segmentos AC, AB e BC são iguais e, portanto, o triângulo ABC construído é equilátero (*Definição 20*).

Para ver a demonstração da Proposição 1 representada na forma de tabela Enunciado/Regra de Dedução ou representada na forma de grafo, veja a tab.4.1 e a fig.4.1. A fig.1 apresenta a construção geométrica (que deve ser feita com régua e compasso) desenvolvida na demonstração.

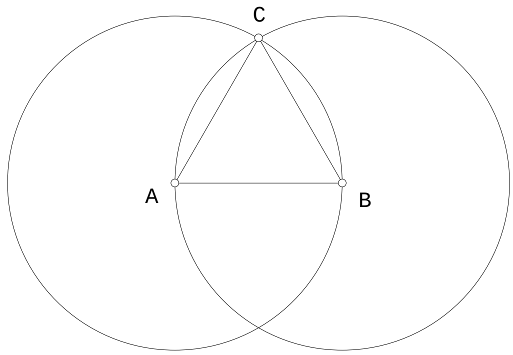

FIGURA 1 - Construção geométrica da Proposição 1

Como visto acima, a primeira proposição independe de qualquer outra proposição. A construção do triângulo é clara e a prova de que ele é um triângulo equilátero é deduzida através de uma seqüência de enunciados logicamente organizados. Esta proposição enquadra-se como um problema de construção com régua e compasso.

Proposição 2: *Colocar um segmento de reta igual a um segmento de reta dado com uma das extremidades em um ponto dado.*

Hipótese: ponto A; segmento BC

Tese: segmento BC = segmento AL

Demonstração: Sejam A o ponto dado e BC o segmento de reta dado. Traçar o segmento AB do ponto A ao ponto B (*Postulado 1*). Construir o triângulo equilátero ABD sobre a reta AB (P*roposição 1*, que uma vez já demonstrada, pode ser utilizada diretamente sem necessidade de repetir passo a passo sua demonstração). Prolongar as retas AE e BF a partir dos segmentos DA e DB (*Postulado 2*). Descrever um círculo C1 de centro B e raio BC (*Postulado 3*). Chamar de G o ponto de interseção do círculo C1 com a reta BF. Descrever um círculo C2 de centro D e raio DG (*Postulado 3*). Chamar de L o ponto de interseção da reta AE com o círculo C2. BC é igual a BG, pois ambos são raios do círculo C1 (*Definição 15*). DL é igual a DG, pois ambos são raios do círculo C2 (*Definição 15*). Como DA é igual a DB (*Definição 20*), então o resto AL é igual ao resto BG (*Axioma 3*). Mas BC é igual a BG (já provado) e AL e BC são iguais a BG. Então AL é igual a BC (*Axioma 1*). Logo, o segmento AL, igual ao segmento BC dado, foi colocado sobre a extremidade dada A. O segmento AL foi construído sobre o ponto A e igual ao segmento BC dado, e a tese foi provada.

A demonstração da Proposição 2 nas representações de tabela Enunciado/Regra de Dedução e grafo pode ser conferida na tab.4.2 e na fig.4.2, respectivamente. A construção geométrica está ilustrada na fig.2.

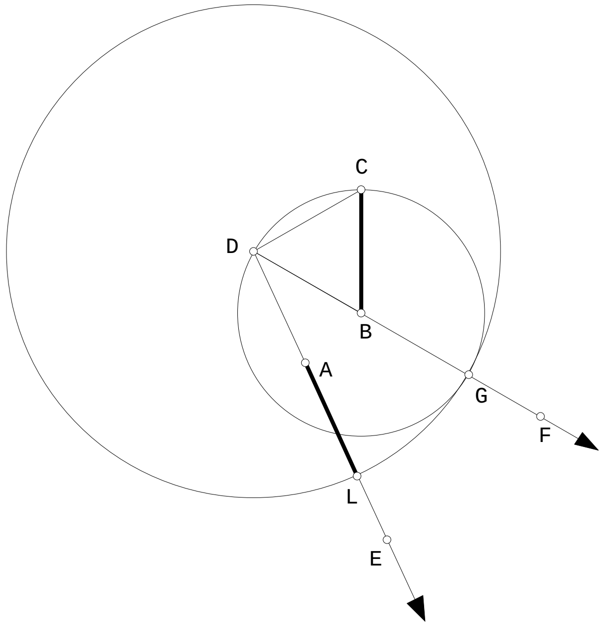

FIGURA 2 - Construção geométrica da Proposição 2

A prova dessa proposição precisou da utilização da proposição anterior para a construção do triângulo equilátero. Dentre os postulados, foi utilizado o Postulado 2, eu não havia sido utilizado na demonstração anterior. Este postulado garantiu a prolongamento de um segmento de reta. Ainda foi utilizado o Axioma 3, que afirma que se igualdades são subtraídas de igualdades, então os restos são iguais.

Esta proposição é novamente um problema de construção com régua e compasso. É uma construção bastante inteligente para se resolver um problema aparentemente simples.

Proposição 3: *Dados dois segmentos desiguais, cortar, do segmento maior, um segmento igual ao segmento menor.*

Hipótese: segmento AB; segmento CD; AB > CD

Tese: segmento AF = segmento BC

Demonstração: Sejam AB e CD os dois segmentos de reta desiguais dados e AB o maior deles. Colocar AE sobre ponto A igual ao segmento CD (*Proposição 2*). Descrever o círculo C1 com centro A e raio AE (*Postulado 3*). Chamar de F o ponto de interseção do círculo com o segmento AB. AF é igual a AE, pois são raios de um mesmo círculo (*Definição 15*). CD é igual a AE (já provado) e, portanto, AF e CD são iguais a AE. Logo AF é igual a CD (A*xioma 1*).

A tab.2 mostra o desenvolvimento da demonstração na estrutura Enunciado/Regra de Dedução.

TABELA 2 - Demonstração da Proposição 3

Enunciado	Regra de Dedução
Segmento AB	Hipótese
Segmento CD	Hipótese
Segmento AB > Segmento CD	Hipótese
Segmento AE = Segmento CD	Proposição 2 (CD, A)
Círculo A-AE	Postulado 3 (A, AE)
Ponto F	Interseção (A-AE, AB)
Segmento AF = Segmento AE	Definição 15 (AF, AE)
Segmento AF = Segmento CD	Axioma 1 (AF, CD, AE)

A fig.3 mostra a construção geométrica desenvolvida na demonstração da Proposição 3.

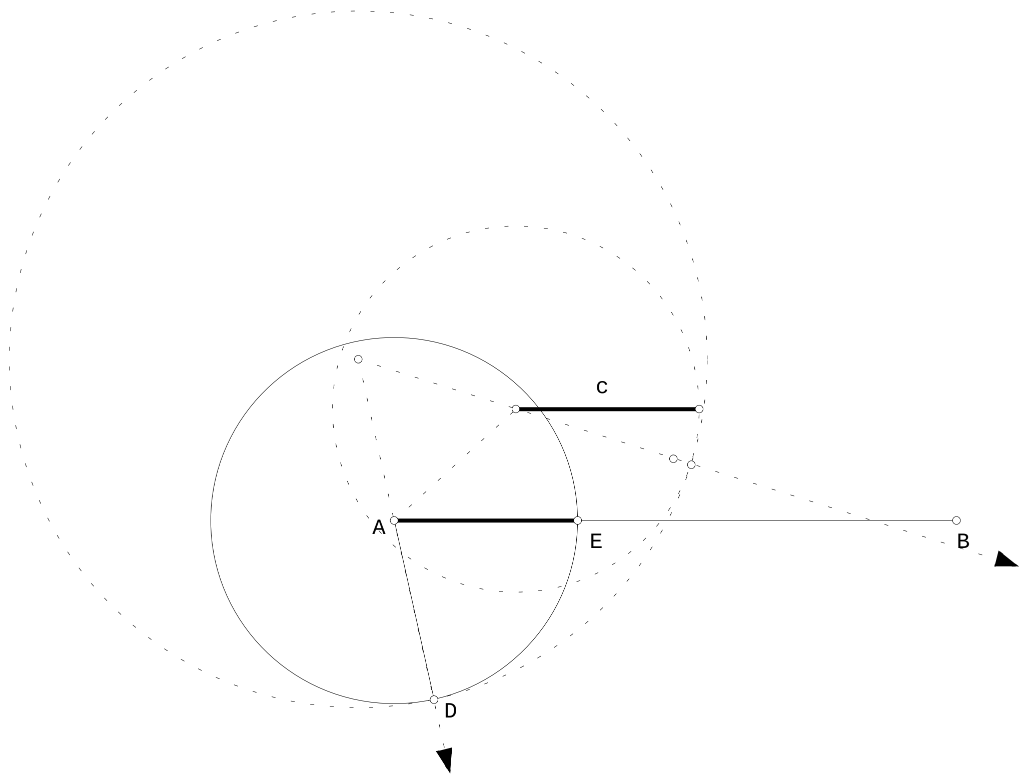

FIGURA 3 - Construção geométrica da Proposição 3

As construções pontilhadas são referentes à construção do segmento AE igual a CD com extremidade no ponto A.

Todas as regras de dedução utilizadas nessa demonstração já foram detalhadas nas demonstrações anteriores, dispensando comentários.

Vale, porém, ressaltar, que a Proposição 3 também é um problema de construção com régua e compasso.

Proposição 4: *Se dois triângulos têm dois lados iguais respectivamente, e têm os ângulos formados por esses lados também iguais, então eles também têm as bases iguais, os triângulos são iguais e os ângulos restantes são iguais respectivamente, chamados de ângulos opostos a lados iguais.*

Hipótese: triângulo ABC; triângulo DEF; AB = AC; DE = DF; $B\hat{A}C = E\hat{D}F$

Tese: BC = EF; $A\hat{B}C = D\hat{E}F$; $A\hat{C}B = D\hat{F}E$

Demonstração: Sejam ABC e DEF os dois triângulos dados. Sejam AB e AC iguais a DE e DF, respectivamente. Seja o ângulo $B\hat{A}C$ igual ao ângulo $E\hat{D}F$. Se o triângulo ABC é superposto sobre o triângulo DEF, e se o ponto A é colocado sobre o ponto D, e se a reta AB é colocada sobre DE, então ponto B coincide com o ponto E, porque AB é igual a DE. AB coincide com DE, a reta AC também coincide com DF, porque o ângulo $B\hat{A}C$ é igual ao ângulo $E\hat{D}F$. Logo, o ponto C coincide com o ponto F, porque AC é igual a DF. Mas B também coincide com E. Logo, a base BC coincide com a base EF e, portanto, são iguais (*Axioma 4*). Então o triângulo ABC coincide com o triângulo DEF e, portanto, ambos são iguais (*Axioma 4*). Os ângulos restantes também coincidem e o ângulos $A\hat{B}C$ é igual ao ângulo $D\hat{E}F$ e o ângulo $A\hat{C}B$ é igual ao ângulo $D\hat{F}E$.

A tab.3 apresenta a demonstração da Proposição 4 representada na forma de tabela Enunciado/Regra de Dedução.

TABELA 3 - Demonstração da Proposição 4

Enunciado	Regra de dedução
Triângulo ABC	Hipótese
Triângulo DEF	Hipótese
Segmento AB = Segmento DE	Hipótese
Segmento AC = Segmento DF	Hipótese
Ângulo BAC = Ângulo EDF	Hipótese
Triângulo ABC superposto sobre Triângulo DEF	Construção
Ponto A coincide com Ponto D	Construção
Ponto B coincide com Ponto E	Afirmações 3 e 7
Segmento AB coincide com Segmento DE	Afirmações 7 e 8
Segmento AC coincide com Segmento DF	Afirmação 5
Ponto C coincide com Ponto F	Afirmações 4 e 10
Segmento BC coincide com Segmento EF	Afirmações 8 e 11
Segmento BC = Segmento EF	Axioma 4 (BC, EF)
Triângulo ABC coincide com Triângulo DEF	Afirmações 9, 10 e 12
Triângulo ABC = Triângulo DEF	Axioma 4 (ABC, DEF)
Ângulo ABC = Ângulo DEF	Afirmação 15
Ângulo ACB = Ângulo DFE	Afirmação 15

A fig.4 ilustra a construção geométrica da Proposição 4.

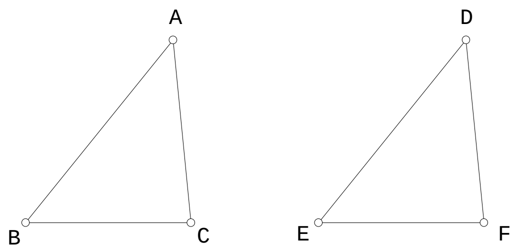

FIGURA 4 - Construção Geométrica da Proposição 4

Proposição 5: *Em um triângulo isósceles, os ângulos da base são iguais e, se os lados iguais são prolongados, então os ângulos abaixo da base são iguais.*

Hipótese: triângulo ABC; AB = AC

Tese: $A\hat{B}C = A\hat{C}B$ **;** $F\hat{B}C = G\hat{C}B$

Demonstração: Seja ABC um triângulo isósceles com o lado AB igual ao lado AC (*Definição 20*). Sejam as retas BD e CE os prolongamentos das retas AB e AC (*Postulado 2*). Tomar um ponto arbitrário F sobre BD. Corta-se AG igual a AF de AE (*Proposição 3*). Construir os segmentos FC e GB (*Postulado 1*). Como AF é igual a AG e AB é igual a AC, portanto os lados FA e AC são iguais aos lados GA e AB, respectivamente, e eles contêm um ângulo comum, o ângulo FAG. Portanto, os segmentos FC e GB são iguais, os triângulos AFC e AGB são iguais e os ângulos restantes ACF e AFC são iguais aos ângulos ABG e AGB, respectivamente (*Proposição 4*). Como AF é igual a AG e AB é igual a AC, então os restos BF e CG são iguais (*Axioma 3*). Como FC é igual a GB, BF é igual a CG, FC é igual a GC, os ângulos BFC e CGB são iguais e a base BC é comum, então os triângulos BFC e CGB são iguais e os ângulos restantes FBC e BCF são iguais aos ângulos GCB e CBG, respectivamente (*Proposição 4*). Como o ângulo ABG é igual ao ângulo ACF, e o ângulo CBG é igual ao ângulo BCF, então os ângulos restantes ABC e ACB são iguais (*Axioma 3*), e eles estão na base do triângulo ABC. E o ângulo FBC já foi provado que é igual ao ângulo GCB, e eles estão abaixo da base.

A representação da demonstração da Proposição 5 está apresentada na tab.4.

TABELA 4 - Demonstração da Proposição 5

Enunciado	Regra de Dedução
triângulo ABC = isósceles	Hipótese
segmento AB = segmento AC	Hipótese
Reta BD	Postulado 2 (AB)
Reta CE	Postulado 2 (AC)
Ponto F	Sobre (BD)
Segmento AF	Postulado 1 (A, F)
Segmento AG	Proposição 3 (AF, AE)
Segmento FC	Postulado 1 (F, C)
Segmento GB	Postulado 1 (G, B)
Ângulo FAC = Ângulo GAB	Comum (FA-AC, GA-AB)
Segmento FC = Segmento GB	Proposição 4 (AFC, AGB)
Triângulo AFC = Triângulo AGB	Proposição 4 (AFC, AGB)
Ângulo ACF = Ângulo ABG	Proposição 4 (AFC, AGB)
Ângulo AFC = Ângulo AGB	Proposição 4 (AFC, AGB)
Segmento BF = Segmento CG	Axioma 3 (BF, CG)
Triângulo BFC = Triângulo CGB	Proposição 4 (BFC, CGB)
Ângulo FBC = Ângulo GCB	Proposição 4 (BFC, CGB)
Ângulo BCF = Ângulo CBG	Proposição 4 (BFC, CGB)
Ângulo ABC = Ângulo ACB	Axioma 3 (ABC, ACB)

A fig.5 mostra os passos construídos para a demonstração da Proposição 5.

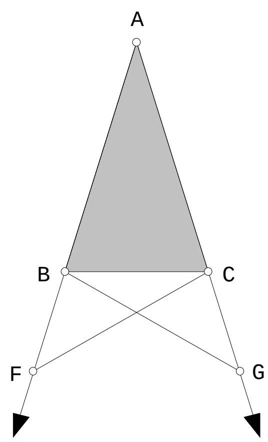

FIGURA 5 - Construção geométrica da Proposição 5

ANEXO 3 Sistemas de Prova Automática em Geometria

A pesquisa e desenvolvimento de programas para a prova de teoremas geométricos atrai a atenção de grande parte da comunidade da Inteligência Artificial. Muitos foram os sistemas já desenvolvidos na área que utilizaram diferentes métodos para sua concepção.

A demonstração automática de teoremas em Geometria tem uma longa história. O primeiro trabalho de peso desenvolvido na área foi realizado por Gelenter [ART 88] e utilizava a idéia de diagrama para forçar a busca de para a prova. Esse programa foi capaz de demonstrar alguns teoremas elementares da Geometria Plana. Era uma abordagem sintética da demonstração automática de teoremas, ou suja, utilizada propriedades de pontos, retas, triângulos, etc. para realizar as provas. Empregava propriedades e definições de conceitos da Geometria Plana para provar os teoremas, usando diagramas e construções geométricas.

A seguir, Tarski desenvolveu uma abordagem para a prova de teoremas em Geometria que não teve repercussão em pesquisas e aplicações da Inteligência Artificial. Utilizou um método de natureza algébrica que, num primeiro momento, pareceu apresentar certa complexidade.

Um melhoramento da teoria desenvolvida por Tarski foi proposta por Collins, que usou a técnica de decomposição algébrica cilíndrica.

Por fim, Wu provocou grande progresso na área de prova automática de teoremas, desenvolvendo um método que verifica se uma conjetura geométrica segue de um conjunto dado de hipóteses. Esse método foi usado com sucesso para demonstrar um grande número de teoremas da Geometria, incluindo até alguns teoremas não triviais.

Tem-se o objetivo de apresentar os sistemas de prova automática em Geometria que já foram desenvolvidos.

1 GEOTHER

GEOTHER (GEOmetry THeorem provER) é um pacote implementado em Maple para manipulação e prova de teoremas geométricos. Seu *kernel* é constituído por seis provadores baseados no método de conjuntos característicos e na base de Gröbner [WAN 96].

O programa apresenta tradução dos teoremas para o Inglês e o desenho geométrico que o representa, além de possuir uma interface gráfica. A fig.1 ilustra a interface do pacote.

Os teoremas geométricos são especificados por meio de predicados. O predicado **`Theorem(H, C, X)`** afirma que H implica em C. O primeiro argumento H é uma lista ou um conjunto de predicados que correspondem à hipótese geométrica e o segundo predicado corresponde à conclusão do teorema. O terceiro argumento é opcional e representa uma lista de variáveis usadas para a computação interna.

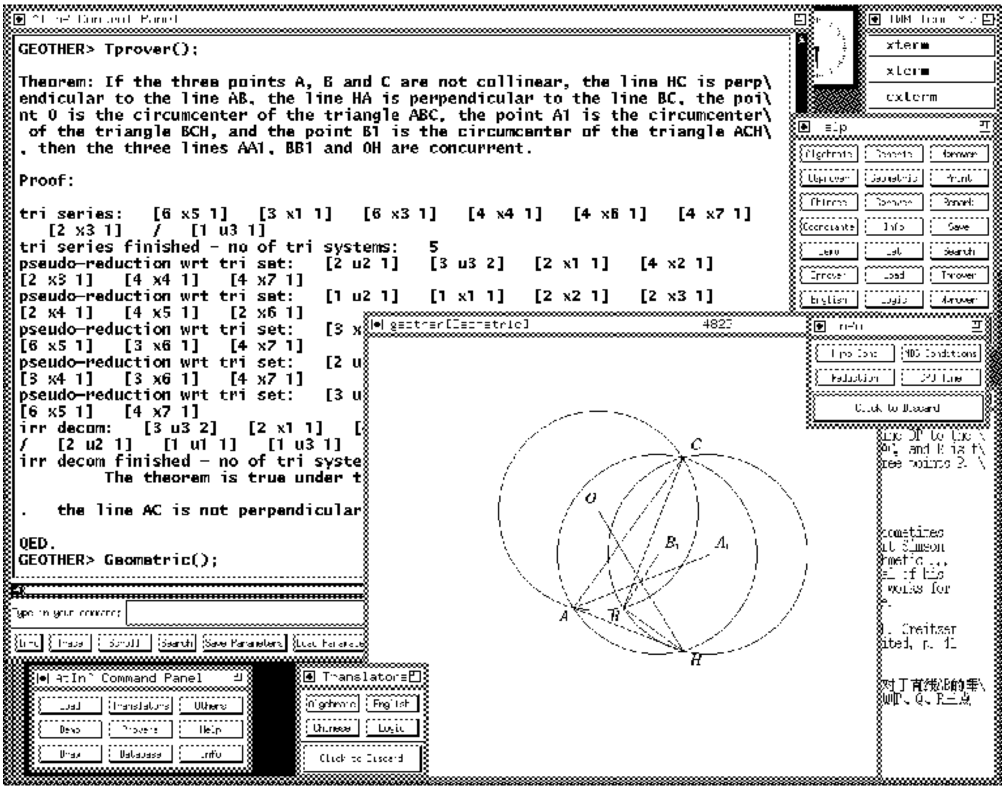

FIGURA 1 - Interface do GEOTHER

2 Geometry Expert

O sistema Geometry Expert (GEX) é um programa para raciocínio geométrico desenvolvido no Departamento de Ciência da Computação da *Wichita State University* [CHO 96].

Suas características são as seguintes: é um provador de teoremas que pode ser usado para produzir centenas de provas de teoremas geométricos, legíveis pelo homem; pode ser usado para derivar novas propriedades ou teoremas; é um editor interativo de diagrama geométrico que pode ser usado para gerar e editar diagramas de afirmações geométricas; é um tutor que pode ser usado para ensinar o usuário como provar teoremas de geometria.

Seu funcionamento é baseado em quatro métodos de prova. São eles: Área, Vetorial, *Full-Angle* e Ponto Fixo. O método da área usa alto nível de lemas geométricos sobre invariantes geométricas tais que a área e a diferença pitagoreana são ferramentas de prova de teoremas geométricos. O método Vetorial é uma variante do método da área e é baseado no cálculo de vetores e números complexos. O método *Full-Angle* é baseado no cálculo de ângulos de 180º. Esse método é baseado em regras e não em procedimentos de decisão. Por fim, o método do Ponto Fixo, que permitiu a construção de um sistema de raciocínio geométrico que é capaz de construir um ponto fixo para determinada configuração geométrica sobre um conjunto de regras ou axiomas geométricos.

A opção de implementar diferentes métodos de prova deve-se à produção de uma variedade de provas com estilos diferentes, que tem grande importância na educação, uma vez que permite aos alunos explorarem as diferentes provas produzidas.

3 GEOMETER

O GEOMETER é um sistema de raciocínio para geometria algébrica, que utiliza uma abordagem refutacional para provar fórmulas em geometria algébrica. O sistema é baseado no método algébrico desenvolvido e implementado por Wu [CYR 88].

O sistema é composto por uma interface gráfica, que utiliza janelas e mouse para especificar propriedades geométricas e a tradução dessas propriedades para a forma algébrica; uma implementação de um algoritmo para desenhar as figuras e assegurar que é a menor configuração que satisfaz as propriedades; e uma implementação do algoritmo de Gröebner usado para provar os teoremas.

A maioria dos problemas geométricos pode ser especificada graficamente no GEOMETER. O usuário pode criar pontos, linhas e círculos.

Sempre que uma nova propriedade é especificada, a figura é adaptada para satisfazer todas as propriedades. Para isso, é usado um método de aproximação numérica baseado na técnica de Newton-Raphson para resolver equações.

Uma vez feita a figura, o usuário pode conjeturar sobre novas propriedades. Para validar suas conjeturas, o usuário pode utilizar o provador de teoremas geométricos. Se o usuário fez conjeturas válidas, retorna *true*. Caso contrário, o usuário pode utilizar a base de Gröebner, que produz informações sobre propriedades adicionais necessárias para assegurar a conjetura. O usuário pode então entrar com essas conjeturas adicionais e utilizar novamente o provador de teoremas.

Um problema de geometria é constituído tipicamente de um conjunto finito de hipóteses que são traduzidas para equações polinomiais; um conjunto finito de propriedades subsidiárias; e uma conjetura da forma de uma equação polinomial a ser provada.

4 Cabri-Euclide

CABRI-EUCLIDE é um ambiente de aprendizagem humana para aprendizagem de provas dedutivas em geometria [LUE 97]. É um micromundo de prova que integra a refutação em sua abordagem.

O sistema permite ao usuário uma interação com um agente racional, que tem o papel de assegurar a coerência do seu raciocínio.

O usuário exprime seu conhecimento através de enunciados (conjeturas, lemas, hipóteses, teoremas, ...). Para um problema de geometria, o usuário pode construir a figura correspondente e produzir os enunciados no micromundo de prova. O agente analisa, então, a estrutura do conjunto de enunciados e diagnostica se há ou não a estrutura de uma prova relativa ao enunciado dado como conclusão. Pode ainda, avaliar um enunciado e sua figura construída e eventualmente refutar, apresentando a figura de um contra-exemplo.

A fig.2 apresenta a interface desse micromundo.

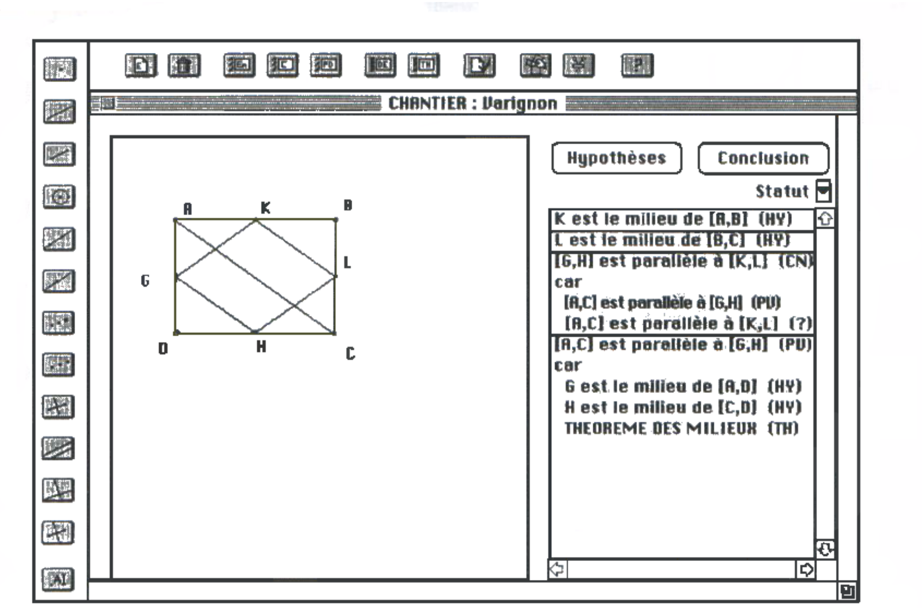

FIGURA 2 - Interface Cabri-Euclide

5 Análise e Comparação dos Sistemas

Observando os sistemas encontrados para prova de teoremas geométricos, pode-se constatar que a verdadeira preocupação desses sistemas é o desenvolvimento de provas automáticas, sem preocupação com o ensino da Geometria Dedutiva.

A maior parte deles utiliza métodos algébricos para construir a demonstração de teoremas, abandonando totalmente a abordagem geométrica desses teoremas.

A tab.1 apresenta algumas características referentes aos sistemas de prova em Geometria e uma comparação entre os sistemas analisados.

TABELA 1 – Comparação de características entre os sistemas

	Prova Automática	Integra figura	Visa o ensino	Métodos algébricos	Diferentes raciocínios	Auxílio de dicas	Linguagem geométrica
Geother	X	X		X			
Geometry Expert	X	X		X			
Geometer	X			X			
Cabri-Euclid		X	X				X

ANEXO 4 Artigo Eurocast 2001

Hyper-Automaton System Applied to Geometry Demonstration Environment

Júlio P. Machado[1], Márcia R. Notare[1], Simone A. Costa[2], Tiarajú A. Diverio[1] and P. Blauth Menezes[1]

[1]Instituto de Informática, UFRGS
Caixa Postal 15064, CEP 91501-970, Porto Alegre - RS - Brasil
{jhapm, notare, diverio, blauth}@inf.ufrgs.br
[2]Centro de Ciências Exatas e Tecnológicas, Unisinos
Av Unisinos 950, CEP 93022-000, São Leopoldo - RS - Brasil
sac@exatas.unisinos.br

Abstract This paper describes the conception and implementation of a learning system on Euclidean Geometry demonstrations and its knowledge base. We use the formalism of finite automata with output to represent and ordain the statements that constitute a geometric demonstration in the knowledge base. The system is built on the MOSCA learning protocol, based on learning with the assistance of examples and interaction among five agents (Mestre, Oráculo, Sonda, Cliente and Aprendiz) involved in the learning process. We briefly revise the Hyper-Automaton concept as a structural model for hypertext and its use as the basis for the central core of the agents in a learning system is analyzed.

1 Introduction

This paper describes the conception and implementation of the knowledge base of a learning system being applied in the Euclidean Geometry context, more specifically to the geometric demonstrations developed by Euclid [1][5], using the formalism of finite automata with output [4] to represent and ordain the statements that constitute a demonstration. The knowledge base is composed of several automata, where the nodes represent the stages of the demonstration and the arcs represent definitions, axioms and postulates that allow the transitions. For the development of this proposal we chose the first of the thirteen books written by Euclid in Elements. This book approaches the Fundamentals of Plain Geometry and is composed by 23 definitions, 5 postulates, 5 axioms e 48 propositions (34 theorems and 14 problems)[5].

An automaton can be represented by a graph structure, which may be manipulated in several ways [8]. Formal semantic graphs may be used to provide a programming interface for the control of hypermedia materials [2], and have been used to good effect in computer-aided instruction and other fields where control over the traversal of a graph is important.

In this sense, this work extends the use of the Hyper-Automaton system, which was designed to manage hyperdocuments in the Web [6][7][9][10], to a model for the knowledge base of an agent-based learning system in Geometry demonstrations.

Initially, the concept of the Hyper-Automaton for hyperdocuments structured as automata is presented and we briefly discuss its use in hypermedia systems. Next, we introduce the multi-agent learning environment on Euclidean Geometry. Then, we define the protocol for the agent´s communication and the knowledge base built over the Hyper-Automaton. Also we present, in detail, an example depicting the information stored in the knowledge base and a possible interaction scenario.

2 Hyper-Automaton

Hyper-Automaton is a system and model which main purpose is to augment the WWW with a hypermedia service based on the application of concepts inherent in Computing Science, specially Automata Theory, Category Theory and Hyperdocuments technology. The system development was aimed at supporting the control of hyperdocuments, mainly for Web-based courses. The system itself consists of three main components: an automata authoring tool, a navigation interface, and a server.

The formal model of hypertext proposed was based on a Finite Automaton with Output [4] representation of a hyperbase structure. It takes advantage of the fact that automata not only capture the descriptive power of Directed Graphs, known to be a useful abstraction in hypertext systems [2], but provide as well a mathematically precise abstract machine for control and analysis of hypertext execution or browsing and is also an universally known formalism. In this sense, automata give us a more computational model for hypermedia technology then directed graphs.

For completeness of our discussion we first provide a short set of definitions for the Hyper-Automaton. The notation style is that commonly used in Automata Theory, similar to [4].

A Deterministic Finite Automaton or Finite Automaton is defined as a 5-tuple $M=(\Sigma,Q,\delta,q_0,F)$ where:

Σ input alphabet, set of symbols from which we may build up words/strings suitable for feeding into the automaton;

Q finite set of possible states;

$\delta: Q\ \Sigma \rightarrow Q$ next-state function, which is a partial function that determines a new state for the automaton based on the current state and the input symbol read;

q_0 initial state, it belongs to Q ;

F set of final states, F is a subset of Q .

The Mealy Machine is represented by a 6-tuple $M=(\Sigma,Q,\delta,q_0,F,\Delta)$ where:

Σ input alphabet;

Q finite set of possible states;

$\delta: Q\ \Sigma \rightarrow Q\ \Delta^*$ next-state function with assigned output, which is a partial function that determines a new state for the automaton and an output based on the current state and the input symbol read;

q_0 initial state;

F set of final states;

Δ output alphabet.

The Moore Machine is analogous and is represented as a 7-tuple $M=(\Sigma,Q,\delta,q_0,F,\Delta,\delta_S)$ where:

Σ input alphabet;

Q finite set of possible states;

δ: Q $\Sigma \rightarrow Q$ next-state function, which is a partial function that determines a new state for the automaton based on the current state and the input symbol read;

q_0 initial state;

F set of final states;

Δ output alphabet;

δ_S: Q $\rightarrow \Delta^*$ next-output function, which is a total function that assigns an output word to the automaton states.

The visual interface of the browser environment provides the user with a tangible interpretation of the Mealy/Moore Machines represented by the Hyper-Automaton.

The automaton n-tuples have counterparts in the structure of hyperdocuments in the Web. The output alphabet Δ is annotated with units of information (hypermedia HTML pages) and, in that case, the result of the next-output function δ_S (Moore) or the next-state function δ (Mealy) is the display of documents (output words Δ^*) in the browser window. The input alphabet Σ that labels the transitions between states in the automaton are displayed as links that can be selected. The link itself is the projection of the next-state function δ in the hypertext environment. If a link is followed, then the current displayed contents are deactivated and the contents mapped to the output nodes (Moore) or transitions (Mealy) are activated, in accordance to the transition executed. One important note is that since we are dealing with hypermedia systems in the Web, the concept of initial and final states becomes a little fuzzy, or in a better sense it becomes an external nondeterminism. All states in the Hyper-Automaton are possibly initial or final. In other words, we may initiate the browsing process from any state and we may stop browsing in any state either.

With the use of Finite Automaton the links are implemented as transition functions and are stored in a matrix representing the source state and destination state, and they are not embedded in the HTML code. Such structure constitutes what is defined as external links and has some advantages: the linked files themselves are not changed by creating hypertext references between them, any file can be edited without altering the linking structure, and, in terms of reuse of hypermedia materials once there is no hard-coded links in the pages it is a straightforward procedure.

For an in depth discussion of the Hyper-Automaton and its relationship with hypermedia, including non-determinism, adaptation mechanisms, modelling of on-line adaptive exercises, and operations for hyperdocuments composition see [7][10][9][6].

3 Learning Environment on Euclidean Geometry

The Learning Environment on Euclidean Geometry - LEEG - is an Euclidean Deductive Geometry's learning environment and its objective is helping in the process of demonstration construction of the Euclidean Plain Geometry.

The system proposes the learning of geometric demonstrations with the assistance of examples or counter-examples, that characterize messages from the system relative inconsistent statements or incorrect terms of Euclidean Geometry.

It follows a multi-agent architecture composed of five different agents: Mestre, Oráculo, Sonda, Cliente, Aprendiz. In a few words, the system works as follows.

Propositions of the Euclid's first book are submitted to Aprendiz to be demonstrated. The demonstrations developed by Aprendiz are constructed on a table Statement/Deduction Rule (see Table 1), in which every step produced by Aprendiz is attended by the system (Mestre agent). The system sends hints to Aprendiz when it identifies incorrectness in the demonstration building process. The messages sent by the system (Oráculo and Sonda) to Aprendiz have the objective to alert about wrong deduction steps or to incentive the continuity of the demonstration. With a stored representation of the automaton knowledge base, it is possible to manage the execution of the individual users' tracks through the analysis of the automaton paths. It is also possible to the users resume their interaction with the system. The users can continue from the point they left during the previous interaction with the learning environment. All we need to do is saving the automaton execution state.

Thus, the user's learning process on Euclidean Geometry is entirely assisted by the LEEG system and its artificial agents (Mestre, Oráculo and Sonda). This three agents have the purpose of accompanying and helping the user's (Aprendiz) knowledge building. The interaction among the several agents is presented in the next section.

Our focus is on using the Hyper-Automaton for organizing information resources to engage students in comprehending, interpreting, and evaluating materials, providing knowledge building and critical thinking.

4 Modeling the Interaction

The learning system for deductions in Geometry is composed of five agents, which are proposed in the learning protocol called MOSCA - Mestre, Oráculo, Sonda, Cliente, Aprendiz - in Portuguese (in English, respectively, Master, Oracle, Probe, Client, Apprentice). It was developed by Reitz [12] and is being used to mould the interaction among the agents involved in the learning process. Briefly, the agents interact among each other in the following way:

- ***Aprendiz:*** receives a proposition from Cliente to be demonstrated. A proposition is composed by a set of hypotheses and a thesis. The hypotheses of the proposition are true statements, which can and must be used in its demonstration. The thesis, however, is the statement that must be proved, through a logic sequence of statements formed by the evident or already demonstrated assertions. The demonstration is deductive axiomatic. It starts from the hypothesis to construct an ordering of statements, which should be equivalent to the structure of the knowledge base. The arcs between the stages of the demonstration are established by analysis of examples received from Oráculo and Sonda agents.

- ***Oráculo:*** in agreement with the Mestre's signaling, the agent interacts with Aprendiz through irrefutable examples, with the purpose of helping Aprendiz to establish the arcs between the stages of the demonstration, formulate and synthesize concepts and relationships between the statements. The examples are stored in a database, which is organized according to the arcs of the knowledge base.
- ***Mestre:*** constantly access the Aprendiz's construction of the demonstration, checking its structure. The verification of the learning is done by comparison between the Aprendiz and environment's knowledge base. This comparison is made step by step, for each new statement included by Aprendiz in the demonstration. When exists an equivalence, we say that the learning is put into effect. Aprendiz is stimulated and Oráculo is signalled to change to the new stage.
- ***Sonda:*** its function is to send examples to Aprendiz, but with refutable solution. The counter-examples can be sent when Mestre perceives that Aprendiz is concluding erroneously an arc in its knowledge structure. Aprendiz must reflect and argue with Mestre to establish the correct arcs.
- ***Cliente:*** starts the process through submission of a proposition to Aprendiz. It has two ways to access the proposition base. The first one consists of an inductive reasoning: it does not submit propositions that require results not demonstrated yet. The second approach concerns a recursive reasoning: propositions that make use of other propositions that were not yet proved by Aprendiz can be submited.

In the LEEG, the agents Mestre, Oráculo and Sonda are artificial and the agents Aprendiz and Cliente are human. The Fig. 1 shows the scheme of communication of the five agents of LEEG.

The system's knowledge base is the complete set of all possible statements that would be used by Aprendiz in every proposition demonstration. Every proposition demonstration is developed at the system's knowledge base and obey the logic sequence of the statements and deduction rules that will be reproduced by Aprendiz. In the LEEG, the knowledge base is composed by 23 definitions, 5 axioms, 5 postulates and 48 propositions demonstrated.

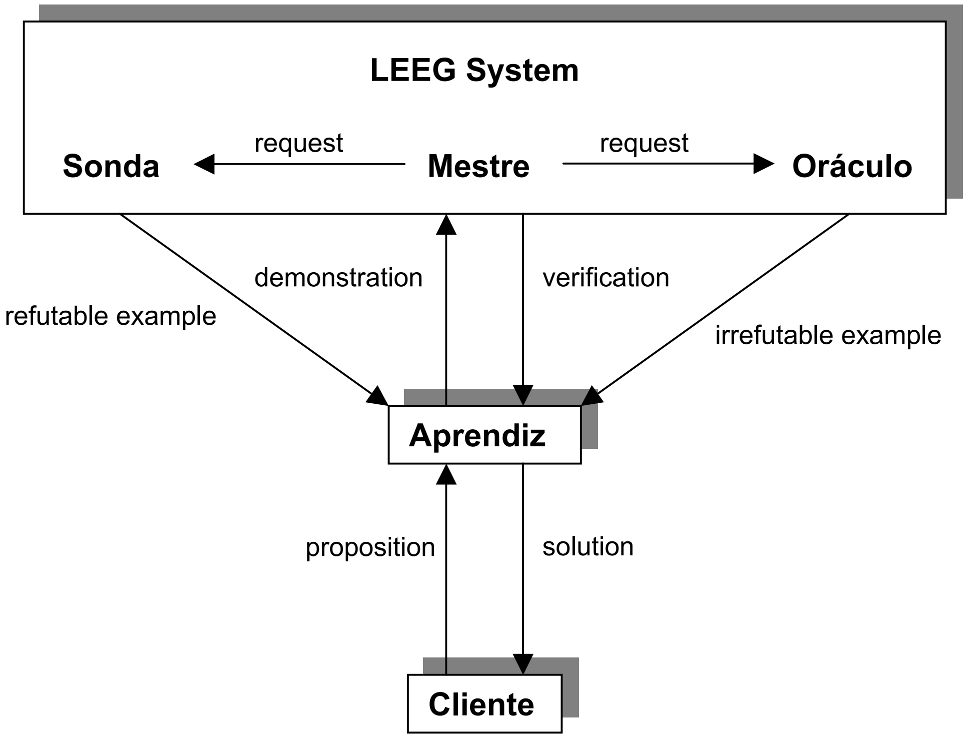

Fig.1. LEEG's scheme of communication

5 Modeling the Knowledge Base

The Euclidean Geometry is a classical example of axiomatic system and its formation structure is deducted from a set of basic premises that descend the others proceeding propositions [11].

Being a logic proposition a sequence of symbols built over well-defined (basic) rules, we define the set Prop as the least set of logical propositions with properties admitted in the construction rules. A sequence $\varphi_0,...,\varphi_n$ is a formation sequence of a logical proposition φ if $\varphi_n = \varphi$ and for all in , φ_i is uniquely determined by the previous propositions, according to the basic construction rules of a logical proposition. A derivation $\subseteq$PropProp is a relation between sets of propositions, denoted $\Gamma\varphi$, i.e., there is a derivation with conclusion φ and all the hypotheses in Γ .

The terms involved in a deductive system are the following [1]:

1. ***Definition***: an assertion that only requires a comprehension of the terms applied;
2. ***Postulates***: principles or facts acknowledged but not demonstrated and admitted without demonstration;
3. ***Axioms***: evident propositions and not subject to demonstration;
4. ***Propositions***: object's property assertions (theorems) or steps or its construction steps (problems) that must be subject to demonstration.

In other words, the definitions, postulates and axioms compose the evident statement set of a deductive axiomatic system, which are acknowledged as true with no need to prove it. The propositions, however, must be proved based on these statements. When a proposition is demonstrated, we can acknowledge it as being true, and use it for the demonstration of other previous propositions, that is, it starts to compose the set of evident statements.

A theorem is a statement of the type $p \to q$ that prove be true ever. The propositions p and q are denominated, respectively, hypothesis and thesis of the theorem. This way, a theorem's demonstration starts with the hypothesis and through application of axioms, definitions and postulates, proves the thesis. This means that, given a theorem, it is fundamental, before starting its demonstration, to identify the hypothesis and the thesis. A demonstration will be a set of statements withdrawn from the definitions, postulates and axioms and the propositions already demonstrated, besides the hypothesis of the proposition, strictly structured and ordained in an hierarchical order. The structure and the demonstration hierarchy must be obeyed so that we can effectively have a deductive axiomatic system.

The deduction process is then a system over a set of propositions, which must be proved. The proof implies a logic sequence over the set of assertions, that corresponds to a derivation tree containing all possible sub-trees that leads to the desired thesis. Trough the logic eyes, a proof would be a derivation tree in which the derivation rules are determined by the statements (basic axioms, postulates, definitions, and propositions already proved), and the sub-trees represent the intermediate paths of deduction. In fact, Godel proved that first order theories with equality is not complete. Similarly, the second order predicate calculus is not complete. As we can see in [3] even though higher order logic does not give us completeness, we can still do formal reasoning to prove the validity of many higher order well formed formulas. Hein [3] presents a familiar example in Euclidean Geometry to see how higher order logic comes into play when we discuss elementary geometry.

Such structure can also be seen as an automaton representing all possible deduction paths that leads to the correct thesis. In this sense, the use and interpretation of the Hyper-Automaton as the main knowledge base for the reasoning on Geometry is quite similar to the use in hyperdocuments. We will return to this point later.

The construction of a deductive demonstration can be associated to two representations: graph and text. The Fig. 2 and the Table 1shows the demonstration of the first Euclid's proposition, constructed over graph an textual representation, respectively. In the following, we present the process of building the knowledge base for the "*Proposition 1: To construct an equilateral triangle on a given finite straight line.*".

In the graph representation, the bold faced nodes (first and last) correspond, respectively, to hypothesis and thesis of the proposition and must be identified in its enunciation. In the proposition demonstration, we start with the hypothesis (accepted as true) and, through a logic sequence of statements, we proved the thesis to be true. Every statement included in Table 1 is strictly justified by axioms, definitions and postulates. The statements that do not present explicit justification are justified by previous statements of the current demonstration.

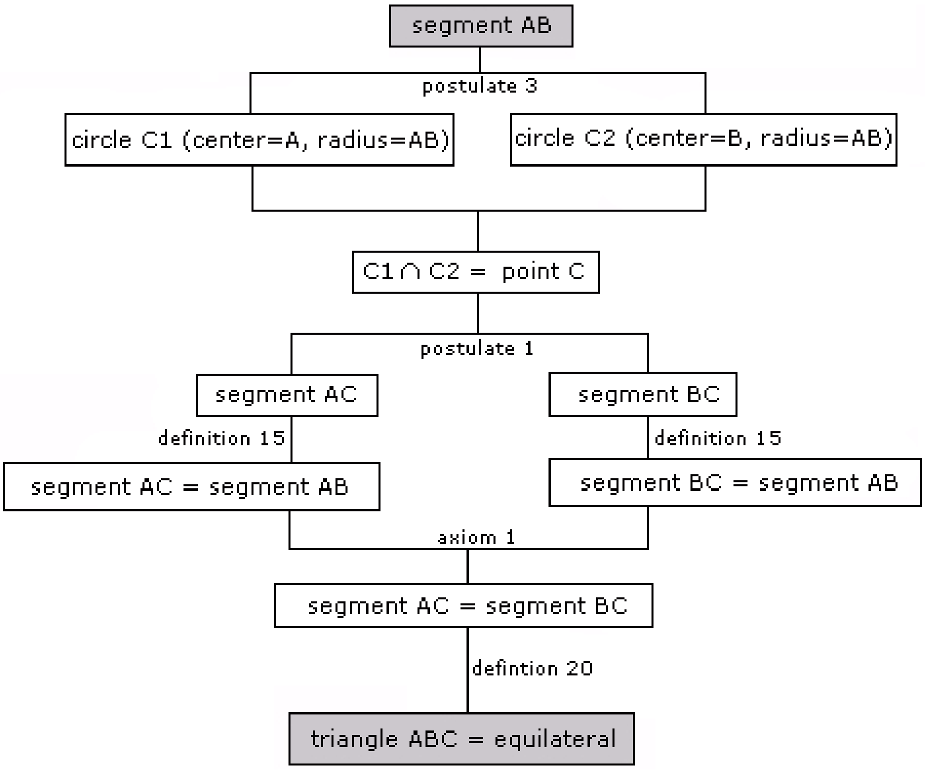

Fig.2. Demonstration of Proposition 1, graph representation

In the textual representation, the deduction rules accepted by LEEG are the 23 definitions, 5 axioms and 5 postulates of the Euclidean Plain Geometry, applied to the correct elements. We take as example of deduction rule the Postulate 1 ("*To draw a straight line from any point to any point.*") in order to illustrate the application. This rule makes reference to the construction of a segment from two points. This means that it must be applied to exactly two points. The first an last statements must be, respectively, the hypothesis and thesis of the proposition, as in the graph representation.

In the LEEG's knowledge base, as previously pointed, the demonstrations structure is implemented through the Hyper-Automaton model, that permits a efficient data organization. The automaton represents all the possible reasoning of deduction that conduct from hypothesis to correct thesis. This means the language accepted by the automaton is a correct deduction in the system. The set of assertions can be seen as the input alphabet for the automaton, labeling the transitions. The set of states represents the current status in the reasoning process: the intermediate objects that are being built to construct the thesis. The next-state function is represented by links between statements, i.e., by each justification that permits the transition from a state to other. The initial state corresponds to proposition's hypothesis and the final state is the proposition's thesis. The output alphabet has two roles

Table 1. Demonstration of Proposition 1, textual representation

Statement	Deduction Rule
segment AB	(Hypothesis)
circle A, AB	Postulate 3 (A, AB)
circle B, AB	Postulate 3 (B, AB)
point C = intersection of two circles	(Statements 2 and 3)
segment AC	Postulate 1 (A, C)
segment BC	Postulate 1 (B, C)
segment AC = segment AB	Definition 15 (AC, AB)
segment BC = segment AB	Definition 15 (BC, AB)
segment AC = segment BC	Axiom 1 (AB, AC, BC)
segment AC = segment AB = segment BC	(Statements 7, 8 and 9)
triangle ABC = equilateral	Definition 20 (AB, AC, BC)

inside the system: in the automata representing all the proposition's proofs, it is the result of applying an assertion; in the automata representing incorrect deduction paths, the output corresponds to critiques depicting the incorrectness. The Fig. 3 shows the proposition 1 structured as a automaton.

We perceive, at the automaton's graph representation, that the demonstration can be developed by ten different reasoning, i.e., ten paths conduct to the final state of the automaton. This occurs because the statements that occur simultaneously in the demonstration in the graph structure, were represented by all the possible sequential orderings in the automaton. This model permits considering different reasonings produced by Aprendiz in the learning process, which are equivalent and accepted as correct by LEEG. The transition function of the respective automaton is represented in Table 2.

6 Concluding Remarks

By analyzing the structure of the knowledge base we observe the finite automata, specially the Hyper-Automaton system, and the MOSCA protocol offer a useful model as the basis for the reasoning system on Geometry. It is also simple to implement and includes a set of efficient and costless algorithms. Even though the agent system is not in the Web, there are some advantages in applying the Hyper-Automaton to the demonstration environment:

- Possibility of representing the knowledge using finite automata with output, which is useful for the design of the Mestre agent.
- Verification of the learning process through comparison of automata substructures.
- Modeling of the interaction between the agents Sonda/Oráculo and Aprendiz by the use of links that activate examples and counter-examples.

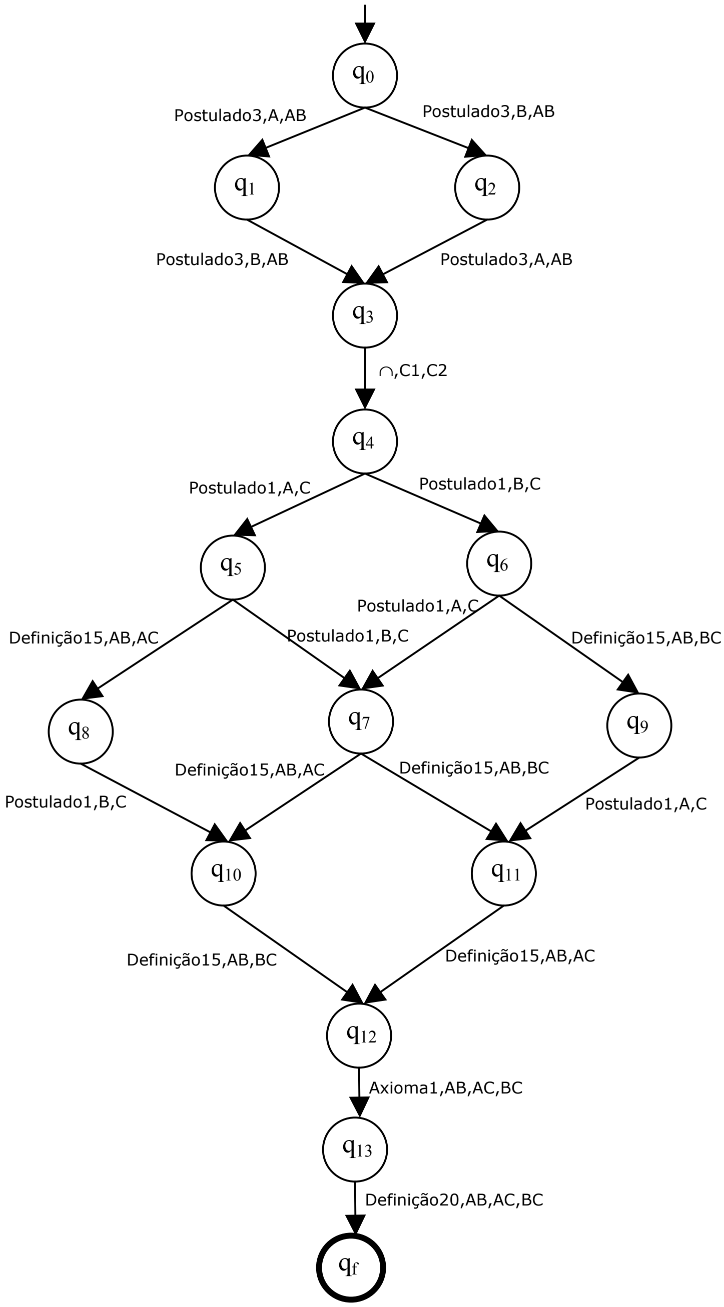

Fig.3. Graph of automaton

Table 2. Transition function of the automaton

δ_1	P3,A,AB	P3,B,AB	C1,C2	P1,A,C	P1,B,C	D15,AB, AC	D15,AB, BC	A1,AB,AC,BC	AB=AC, AB=BC, AC=BC	D20,AB, AC,BC
q_0	q_1	q_2								
q_1		q_3								
q_2	q_3									
q_3			q_4							
q_4				q_5	q_6					
q_5					q_7	q_8				
q_6				q_7			q_9			
q_7						q_{10}	q_{11}			
q_8					q_{10}					
q_9				q_{11}						
q_{10}							q_{12}			
q_{11}						q_{12}				
q_{12}								q_{13}		
q_{13}									q_{14}	
q_{14}										q_f

- Modeling propositions as transitions (links in the Hyper-Automaton) between hypothesis and thesis. In case a certain proposition has not yet being demonstrated, its associated transition becomes unavailable until further demonstration.
- They allow a certain computation redundancy that models the different ways of reasoning. As a consequence, we are able to study the Aprendiz agent reasoning process in its diversity in the demonstration of propositions. Such diversity may be modeled as concurrent events using formalisms for concurrency (as CCS), which is planned for future works.

Bibliography

1. Euclid, (trad.)Thomas, H.: The Thirteen Books of Euclid's Elements. Dover Publications, New York (1978)
2. Garzotto, F., Schwabe, D., Paolini, P.: HDM: A Model Based Approach to Hypertext Application Design. ACM Transactions on Information Systems, January 1993, Vol.11, Num.1. (1993) 1–26
3. Hein, J. L.: Discrete Structures, Logic and Computability. Jones and Bartlett, Boston (1995)
4. Hopcroft, J. E., Ullman, J. D.: Introduction to Automata Theory, Languages and Computation. Addison-Wesley (1979)
5. Joyce, D. E.: Euclid's Elements. Clark University, USA (1998) URL http://aleph0.clarku.edu/ djoyce/java/elements/elements.html
6. Machado, J. P., Penczek, L., Morais, C. T. Q., Menezes, P. B.: Finite Automata: a formalism for web courses (in Portuguese, Autômatos Finitos: um formalismo para cursos na web). In: Nunes, D. J., Camargo, M. S.(eds.): 13 Brazilian Symposium on Software Engineering (in Portuguese, 13 Simpósio Brasileiro de Engenharia de Software). SBC, UFSC, Florianópolis, Brazil (1999) 213–223

7. Machado, J. P., Morais C. T. Q., Menezes, P. B., Reis, R.: Structuring Web Course Pages as Automata: revising concepts. In: Mariani, J., Harman, D.(eds.): RIAO 2000, 6 Conference on Content-based Multimedia Information Access, Vol.1. CID, CASIS, Paris (2000) 150–159
8. Menezes, P. B., Sernadas, A., Costa, J. F.: Nonsequential Automata Semantics for a Concurrent Object-Based Language. In: Cleaveland, R., Mislove, M., Mulry, P.(eds.): First US-Brazil Workshop on Formal Foundations of Software Systems, ENTCS, Vol.14. Elsevier Science, Amsterdam (1998) 29 pp.
URL http://www.elsevier.nl/locate/entcs/volume14.html
9. Menezes, P. B., Machado, J. P.: Hyper-Automaton: Hypertext Framework with Categorial Operations. In: Borba, P. H. M., Moura, H. P., Santos, A. L. M.(eds.): 4 Brazilian Symposium on Programming Languages. SBC, UFPE, Recife, Brazil (2000) 29–47 (to appear as an ENTCS volume)
10. Menezes, P. B., Machado, J. P.: Adaptative Web Courses: a categorical framework. In: CASYS'2000, Computing Anticipatory Systems, Dubois, D. (ed.): International Journal of Computing Anticipatory Systems. CHAOS, Liège (2001) (to appear)
11. Meyer, B.: What is Mathematics? In: An Introduction to Axiomatic Systems. Prindle, Weber & Schmidt, Boston (1974) chap.1
12. Reitz, P.: Contribution à l'étude des environnements d'apprentissage: Conceptualisation, Spécifications et Prototypage. Thèse de Doctorat. Université des Sciences et Techniques du Languedoc Montpellier II, Montpellier (1992)

ANEXO 5 Artigo CASYS'2001

Knowledge Anticipation on Agents Relationship in the Geometry Proof System

Márcia R. Notare, Júlio P. Machado, Tiarajú A. Diverio, P. Blauth Menezes

Instituto de Informática and PPGC/UFRGS
Caixa Postal 15064 91501-970 Porto Alegre - RS - Brazil
E-mail: {notare, jhapm, diverio, blauth}@inf.ufrgs.br

Abstract
This paper describes the Geometry demonstration learning system LEEG (Learning Environment on Euclidean Geometry). This system was constructed on a learning environment composed of five agents that interact to promote the knowledge construction and evolution. The five agents are: *Mestre*, *Oráculo*, *Sonda*, *Cliente* and *Aprendiz* (or in English, respectively, Master, Oracle, Probe, Client, Apprentice), each one with distinctive and specific behavior. The focus of this work will be the specification of the *Mestre-Oráculo* and *Mestre-Sonda* relationships and the knowledge base specification.

Keywords: knowledge anticipation, learning system, automata theory, *MOSCA* protocol.

1 Introduction

This paper describes the Geometry demonstration learning system LEEG (Learning Environment on Euclidean Geometry). This system was constructed on a learning environment composed of five agents that interact to promote the knowledge construction and evolution. The five agents are: *Mestre*, *Oráculo*, *Sonda*, *Cliente* and *Aprendiz* (or in English, respectively, Master, Oracle, Probe, Client, Apprentice), each one with distinctive and specific behavior.

Cliente starts the learning process sending a proposition to *Aprendiz* to be demonstrated and awaits its solution. *Aprendiz* must construct the demonstration with help of examples/counter-examples sent by *Oráculo/Sonda*. *Mestre* controls the learning process by permanently accessing the knowledge base of *Aprendiz* and comparing to the knowledge base that contains all the correct demonstrations. This means that the agent *Mestre* knows previously the demonstration structure that was proposed to *Aprendiz*. In this way, it always signalizes to *Oráculo* and *Sonda* when it identifies an incoherence in the *Aprendiz* construction, in order to coordinate the sending of examples and counter-examples that help the correct demonstration construction.

Mestre has access to the complete demonstrations proposed by *Cliente*. So, it is, in some sense, able to anticipate the possible incorrectness committed by *Aprendiz*. Thus, it can require that examples have to be send to *Aprendiz*, avoiding the execution of this incorrectness. This permits a better control over the demonstrations construction developed by *Aprendiz*.

A demonstration on Euclidean Geometry is composed by a set of statements that must obey a hierarchical order and must be rigorously justified. By this reason, the knowledge base implementation was structured by Hyper-Automaton model [2] [3] [5]. This model enables an adequate structure for the demonstrations statements and possibility the verification of the learning process through comparison of automata substructures.

The focus of this work will be the specification of the *Mestre-Oráculo* and *Mestre-Sonda* relationships and the knowledge base specification. Also, initially, the anticipatory system attribute had not been specifically tackled in the project, but implicitly incorporated, which motivated us to make it explicit and then explore it in this work.

2 MOSCA Protocol

Formal theories of learning state that a minimal learning environment should comprise a learner in communication with an oracle in order to enable learning.

Based on these assumptions, Reitz [4] has proposed a learning environment known as MOSCA (**M**estre + **O**ráculo + **S**onda + **C**liente + **A**prendiz) as shown in Figure 1. This environment is composed by 5 distinct entities (considered as human or artificial agents), each with specific behavior, according to proposed aims and interacting in the learning process. MOSCA is a learning protocol based on learning by example, which have been used and adapted by many in the literature [7] [8] [9].

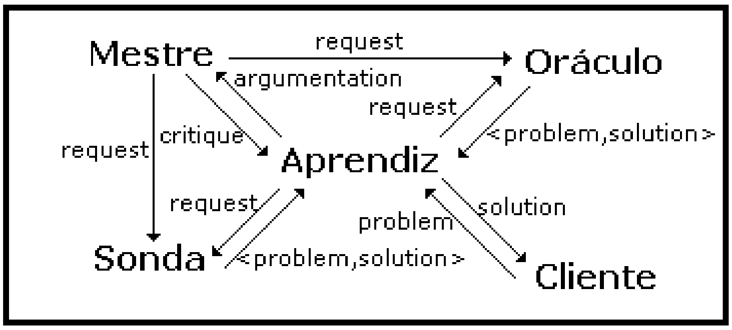

FIGURE 1 – MOSCA learning protocol

In summary, the initial proposal of the MOSCA protocol is as follows: the process starts with *Cliente* which submits a problem to be solved to the *Aprendiz*. The *Aprendiz* also receives problems solved by *Sonda*. However such solutions can be incorrect. The *Aprendiz* compares its solution with *Sonda*'s solution and defends its solution to the *Mestre*. For each argumentation produced by the *Aprendiz*, a critical argument is sent by the *Mestre*. Criticized arguments are memorized by the *Aprendiz*. Every negative argument leads the *Aprendiz* to present a new argumentation.

Initially, the anticipatory system attribute [1] had not been specifically tackled in the project, but implicitly incorporated, which motivated us to make it explicit and then explore it in this work. The *Mestre*, entity responsible for monitoring the actions of the *Aprendiz*, knows in advance the final and partial objectives that the *Aprendiz* should meet. In that way, the *Mestre* can in anticipation, warns the *Oráculo* and the *Sonda* what intervention are necessary in order to reach the final objective successfully. I.e., the *Mestre*, having a global view of the process can lead the *Aprendiz* to the correct learning path, aided by the *Oráculo* and the *Sonda*.

An example of this behavior can be observed in the application of the learning protocol in the context of arithmetic, specifically, in learning how to solve numerical expressions. Next, possible ways that can be followed by the *Aprendiz* in the solution of numerical expressions (Figure 2).

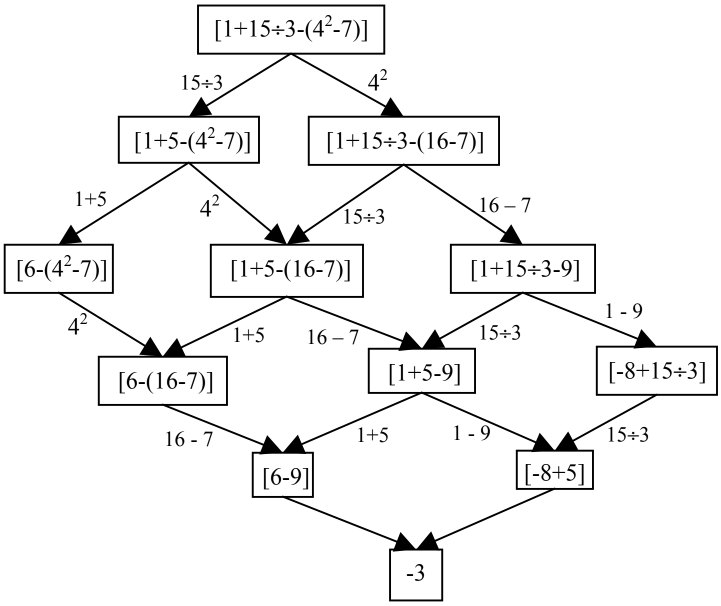

FIGURE 2 - Resolution of arithmetic expression

For each incorrect operation that the *Aprendiz* tries to execute, the *Oráculo* and *Sonda* agents intervene in the construction through messages, with the aim of telling the *Aprendiz* to repeat the operation. For instance, if in the first moment the *Aprendiz* calculate *1+15*, the *Sonda* agent intervenes, warning that division has priority over addition. And this should happen in all operations not available for resolution.

3 Learning Environment on Euclidean Geometry

The LEEG (Learning Environment on Euclidean Geometry) [11] is a learning environment for Deductive Euclidean Geometry, based on adaptations of the MOSCA learning protocol with the aim of helping the construction of theorem proving process in Euclidean Plane Geometry.

The system proposes the learning of geometric proofs constructions with the aid of examples and counterexamples, which characterize interventions of the system when inconsistent statements or incorrect use of terms are used.

As Euclidean Geometry is a classical example of an axiomatic system [10], its structural form can be proved from a certain number of base premises that give rise to derived propositions [6].

The terms involved in a deductive system are the following [10]:

- ***Definitions***: assertion that only requires a comprehension of the terms applied;
- ***Postulates***: principles or facts acknowledged but not demonstrated and admitted without demonstration;
- ***Axioms***: evident propositions and not subject to demonstration;
- ***Propositions***: object's property assertions (theorems) or steps or its construction steps (problems), that must be subject to demonstration.

In other words, the definitions, postulates and axioms make up the set of evident terms in an axiomatic system, which are considered as true without proof. The propositions must be proved from the basic terms and the rules that establish the system. Whenever a proposition is proved, we can admit it as true and use it in order to prove new propositions, i.e., this new proposition is now part of the set of true statements of the system.

In a proposition, the hypothesis and the thesis must be identified. In the case of a deductive proof, the proposition hypothesis is taken as true and should be used in the proof construction. The thesis is the statement of what should be proved, by a rigid logical sequence of statements, composed by evident statements and proved propositions.

A proof then will be a set of mathematical statements that should be justified by the use of definitions, postulates and axioms in addition to the propositions already shown previously. These statements must be rigorously structured and ordered in a logical and hierarchical form. The demonstrations are rigidly structured and this structure must be followed in order to produce a proof within a axiomatic deductive system.

Thus, the LEEG system aims at helping an agent (*Aprendiz*), human or virtual, in the construction of proofs in Euclidean Plane Geometry, have as a basis the rigidity of the axiomatic system. The construction of the demonstrations developed by the *Aprendiz* must follow a logical sequence of construction. Each step of the *Aprendiz* is followed by the *Mestre* agent which identifies inconsistencies and mistakes and activates the message passing mechanism (examples and counter-examples) that intervene in the construction.

The deductive demonstrations in LEEG are developed in textual form, organized in a table composed by two columns, named *Statement* and *Deduction Rule*. The *Aprendiz* will develop its demonstration over them. For each field in the statement column there will be an equivalent field in the deduction rule column, which must be exactly filled by the deduction rule that concluded the corresponding statement. Only statements which cannot be deduced via deduction rules can have the field not filled in.

The LEEG accepted deduction rules comprise 23 definitions, 5 axioms and 5 postulates of Euclidean Geometry, applied to the right elements. For instance, Postulate 1 which states that *"It is possible to draw a straight line from any point to any point"* refers to the construction of a segment from two given points. This means it should be applied to exactly two points.

The first and last statements of the proof must be, respectively, the hypothesis and thesis of the proposition.

Table 1 presents the proof of Proposition 1 *(To construct an equilateral triangle over a give line segment)* as it should be constructed by the *Aprendiz* in the LEEG interface.

TABLE 1 – Demonstration of Proposition 1

Statement	**Deduction Rule**
segment AB	(Hypothesis)
circle A, AB	Postulate 3 (A, AB)
circle B, AB	Postulate 3 (B, AB)
point C = intersection (circle A,AB; circle B,AB)	(Statements 2 and 3)
segment AC	Postulate 1 (A, C)
segment BC	Postulate 1 (B, C)
segment AC = segment AB	Definition 15 (AC, AB)
segment BC = segment AB	Definition 15 (BC, AB)
segment AC = segment BC	Axiom 1 (AB, AC, BC)
segment AC = segment AB = segment BC	(Statements 7, 8 and 9)
triangle ABC = equilateral	Definition 20 (AB, AC, BC)

The fields in the Deduction Rule column that are shown between round brackets do not need to be included in the demonstration developed in the system. The hypothesis falls into this case as it is not a statement deduced from a deduction rule (is extracted from the proposition) and the other statements that fall into this case, in the above example, are only facts with no deduction rules.

Each reasoning step of the *Aprendiz*, i.e., each line in the proof table is supervised by the *Mestre*. The next step is enabled only if the current reasoning is correct. Otherwise, the *Oráculo* and *Sonda* agents are warned in order to send examples and counterexamples that tell the *Aprendiz* about its mistake and make it possible for it to correct the mistake.

In this way, the permanent monitoring of the proof construction developed by the *Aprendiz* allows the *Mestre* to anticipate likely mistakes that could be made by the *Aprendiz*, warning the *Oráculo* and *Sonda* agents to send messages that influence the construction of a correct proof.

Figure 3 shows the MOSCA protocol adapted for the LEEG system.

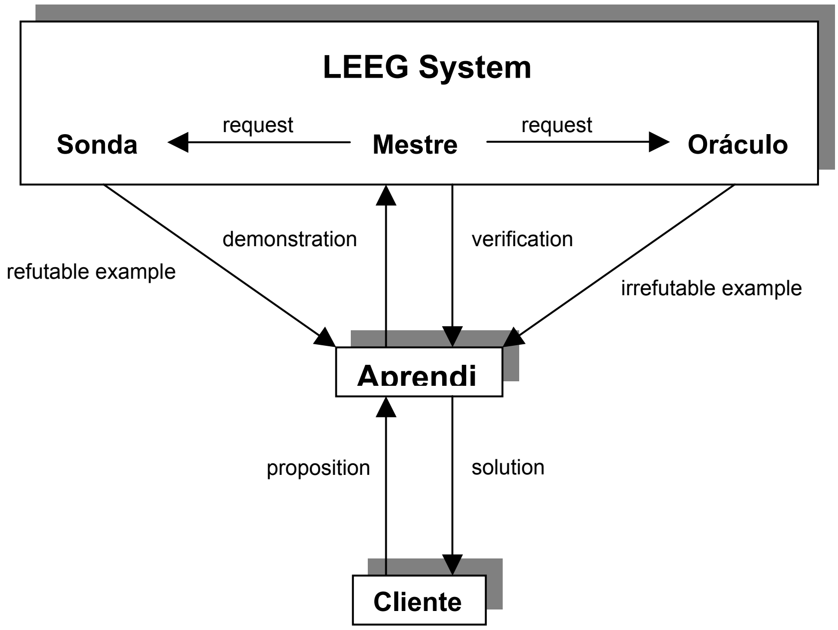

FIGURE 3 – LEEG's version of MOSCA protocol

In the following items, we describe the knowledge base of the system and the interaction between agents while learning how to make proofs about Geometry.

4 Agents Relationship

This section aims at specifying the interaction between the five agents that compose the LEEG system.

4.1 Cliente ↔ Aprendiz Interaction

The communication between *Cliente* and *Aprendiz* agents establishes the beginning of the learning process. The *Cliente* sends a proposition to the *Aprendiz* (Cliente → Aprendiz), from the propositions available for a proof, and awaits the result.

In the chosen proposition, the thesis and hypothesis are identified. The hypothesis should be used by the *Aprendiz* as the initial statement of the proof. The thesis should be proved by a sequence of statements logically deduced and should be the last statement of the proof.

Once the proof is concluded, the *Aprendiz* returns the result of the proof to the *Cliente* (Cliente ← Aprendiz), presenting a complete demonstration.

4.2 Mestre ↔ Aprendiz Interaction

The interaction between the *Mestre* and *Aprendiz* agents is related to the construction (Mestre ← Aprendiz) and verification (Mestre → Aprendiz) of learning. Each state of the proof developed by the *Aprendiz* is sent to the *Mestre* which verifies the construction. Three situations may happen: correct construction, wrong construction of a statement or wrong construction of a deduction rule.

If the construction is correct, the *Mestre* send a signal to the *Aprendiz*, authorizing the next step of the construction.

If the construction of the statement or of the deduction rule is wrong, the *Mestre* do not authorizes the next step of the construction and send a message to the *Oráculo* and *Sonda* agents, which will warn the *Aprendiz*.

The verification of the steps developed by the *Aprendiz* is done though comparisons between its demonstration and the demonstration in the knowledge base of the system. As the *Mestre* has access to the correct construction in the knowledge base, it can anticipate likely mistakes in the construction of a proof by the *Aprendiz*, and prevent them by sending messages about correct steps in the demonstration.

4.3 Mestre → Oráculo Interaction

The communication between the *Mestre* and *Oráculo* agents is made in only one direction and it is related to the request by the *Mestre* to the *Oráculo* of messages to be sent to the *Aprendiz*. This warning happens in two situations: wrong construction of a deduction rule and correct construction.

If the *Mestre* identifies a mistake in the deduction rule constructed by the *Aprendiz*, then it warns the *Oráculo* so that it sends a message to the *Aprendiz* to correct the referred state of the proof.

On the other hand, if the *Mestre* checks the current state of the proof by the *Aprendiz*, and after the habilitation signal no action by the *Aprendiz* is identified by the *Mestre*, a signal is send to the *Oráculo* so that it can send a message encouraging the correct construction. Again, an anticipation about the next state of the proof happens and it helps the *Aprendiz* to construct a correct step.

4.4 Mestre → Sonda Interaction

This interaction is also unidirectional. The communication is established through a signal sent from the *Mestre* to the *Sonda* asking for a reflection message to be sent to the *Aprendiz*.

This signal is sent whenever the *Mestre* identifies an error in the statement of the *Aprendiz* construction. The reflection messages have the aim of helping the *Aprendiz* to identify and correct its error, but provide specific information about the correct construction of the statement.

4.5 Oráculo → Aprendiz Interaction

The communication established between the *Oráculo* and *Aprendiz* agents is related with the examples used to help in the learning process. The *Oráculo* aims to send a message to the *Aprendiz* that helps it in choosing the right deduction rules, for each step of the proof (correction messages) or messages that encourage the *Aprendiz* the continuation of the proof when the *Aprendiz* abandons a proof (incentive messages).

The *Oráculo* has access to a Module of Irrefutable Examples (which is in the Examples Base) which can be accessed depending on the current state of the proof by a signal from the *Mestre*.

4.6 Sonda → Aprendiz Interaction

The *Sonda* agent aims at sending reflection messages to the *Aprendiz* agent, whenever the *Mestre* identifies mistakes in the construction of statements. These reflection messages are determined according to the *Mestre* signal, which identifies the current state of the proof and the mistake made by the *Aprendiz*.

The *Sonda* agent has access to a Module of Refutable Examples (also included in the Examples Base), where each message is associated to a state in the proof.

5 Knowledge Data Structure

Data in the LEEG system are structured as follows: the *Knowledge Base*, including definitions, axioms, postulates and proved propositions; the *Propositions Base* which contains propositions to be demonstrated; the *Examples Base*, composed by two Modules (*Irrefutable Examples* and *Refutable Examples*). Figure 4 presents the data structure of the system.

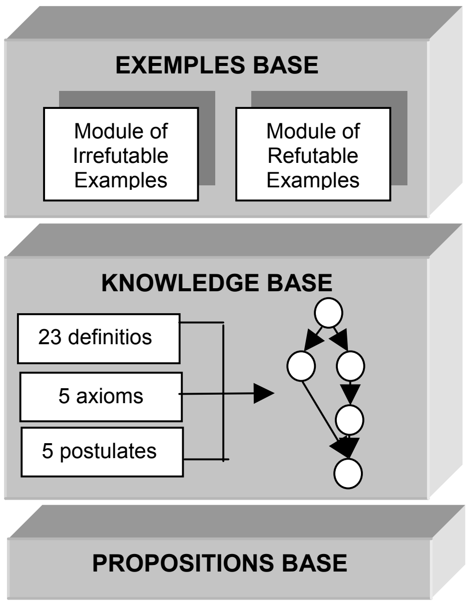

FIGURE 4 – LEEG's data structure

5.1 Knowledge Base

The knowledge base of the system is the complete set of all possible statements that can be used by the *Aprendiz* in the proof of each one of the propositions. The proofs of the propositions are developed in the basis of the system, obeying the logical sequence of the statements and justifications that should be reproduced by the *Aprendiz*. In the case of LEEG, the knowledge base may be formed by 23 definitions, 5 axioms, 5 postulates and 48 proposition theses.

The structure of a proof can be represented by a graph, where all the statements of Geometry being used in the proof are hierarchically organized and the edges between statements are the terms that justify their use (the deduction rules).

In the LEEG's knowledge base the structure of the proofs is organized and implemented by a Finite Automata model, which allows an efficient data organization [2] [3] [5].

5.2 Propositions Base

The propositions base or possible propositions databank which are to be submitted to the *Aprendiz* is determined by the scope of the knowledge base. In the case of Euclidean Plane Geometry, there are 48 possible propositions to be submitted by the *Cliente* to the *Aprendiz*. The *Cliente* agent is responsible for the access to this set of propositions, and chooses a proposition to be submitted to the *Aprendiz* to be proven. The proposition can be chosen *randomly* or *structurally*.

In the latter case the dependency order between propositions is followed, i.e., a proposition will be available for submission if the other propositions needed for its proof were already proved by the *Aprendiz*. In that way, knowledge acquisition is done step by step, in an *inductive* way.

5.3 Examples Base

The examples base of the system is composed by the set of message interventions send by the system (though the *Oráculo* and *Sonda* agents) to the *Aprendiz*.

The interventions send by the system to the Aprendiz have two aims:

- broadcasting of examples containing enough hints on deduction rules, in order to lead the *Aprendiz* agent to a correct proof construction, i.e., making it possible the inclusion of deduction rules consistent with the current proof;
- broadcasting of counter-examples, whenever the *Aprendiz* agent makes a statement mistake, in order to warn it about the mistake make and helping in the correction.

The errors make by the *Aprendiz* can be related to the statements or to the deduction rules. *Incoherence* of a demonstration is a consequence of statements derived from incorrect deduction rules and the *inconsistency* of a demonstration is the introduction of statements with no application of a deduction rule.

The *Mestre* agent is responsible for deciding whether to broadcast messages since it follows and checks, step by step, the reasoning developed by the *Aprendiz* agent. It is by the *Mestre* request that the *Oráculo* and *Sonda* agents lead the *Aprendiz* to the correct construction.

In that way, the *Oráculo* helps the *Aprendiz*, by a *Mestre* signal, by sending examples that lead to the correct deduction rule. On the other hand, the *Sonda* agent is warned by the *Mestre* whenever the *Aprendiz* makes a mistake in the statement of a proof, so that *Sonda* sends a counter-example to warn the *Aprendiz* about its mistake and allows a reflection about it and then a correction.

5.3.1 Irrefutable Examples Module

The irrefutable examples module is accessed only by the *Oráculo*, which is responsible for sending messages containing examples and hints. The messages in this module are irrefutable, i.e., they are correct and can be accepted by the *Aprendiz* with total confidence.

The messages in this module are divided into two categories, identified by their functionality: *Correction Messages* and *Incentive Messages*.

Correction messages are sent whenever an identification error in the *Aprendiz* deduction rules is detected and aims at providing hints to warn the *Aprendiz* about the error and to direct a correction. Figure 5 shows the interaction between the *Aprendiz*, *Mestre* and *Oráculo* agents during the error identification process and the correction message production.

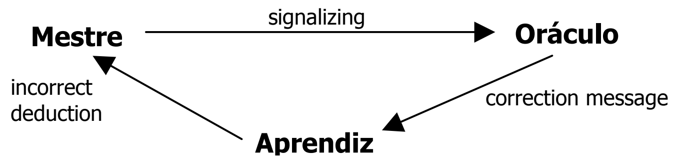

FIGURE 5 – Production of correction message

In summary, the *Aprendiz* produces a wrong deduction rule, the *Mestre* identifies the error by comparing it to the LEEG knowledge base and tells the *Oráculo* to send a correction message about the current state of the proof to the *Aprendiz*. These messages can still be classified into four distinct levels, according to the error identified:

- *hint about the type of element that the deduction rule should be applied* – sent whenever the *Aprendiz* makes a mistake related to the type of geometric element (point, segment,...) on which the deduction rule should be applied. I.e., the *Aprendiz* chooses the right deduction rule, but not the right element. (Example message: *"Postulate 3 must be applied to a point and a segment"*).
- *hint that specifies the elements over which a deduction rule must be applied* - sent whenever the *Aprendiz* makes a mistake in the element(s) (and not in the type of element), specifying which element(s) that the rule should be applied to. (Example message: *"Apply postulate 3 to point A and segment AB"*).
- *message warning about incoherence between statement and deduction rule* - sent to warn the *Aprendiz* that the inserted deduction rule is incoherent with the respective statement, i.e., the statement cannot be deduced from the rule. (Example message: *"Deduction rule incoherent with statement"*).
- *message warning about inconsistency in deduction* - sent whenever the *Aprendiz* does not insert the deduction rule for a statement that should be deduced from a rule. (Example message: *"Inconsistent Deduction"*).

Incentive Messages are sent when two facts are detected: the statement and corresponding deduction rule are correct, but no action from the *Aprendiz* for a certain amount of time (usually set to 3 minutes) is detected, which is interpreted as undecided behavior or withdrawal from learning. These messages aim at stimulating the *Aprendiz* to continue with the proof when indecision or detachment are detected, by sending hints about the next state of the demonstration. Figure 6 presents the interaction between the *Aprendiz*, *Mestre* and *Oráculo* agents during the incentive message production process.

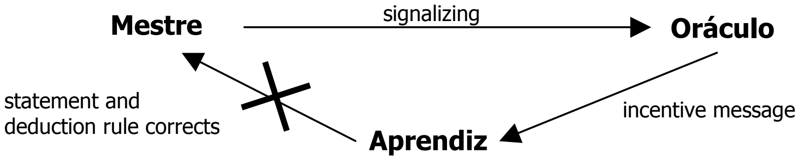

FIGURE 6 – Production of incentive message

Each state of the proofs may have more than one associated incentive message, anticipating the different reasoning strategies that can be adopted by the *Aprendiz*. An example message in this category is the following: *"You can check the equality of segments AB and AC"*.

5.3.2 Refutable Examples Module

The refutable examples module is accessed only by the *Sonda* agent, which sends counter-examples that may be refuted by the *Aprendiz*.

The messages in this module, known as *Reflection Messages*, are sent only in situations in which the *Aprendiz* makes a statement error. These messages aims to warn the *Aprendiz* about the error, creating situations which make him think about the concluded statement. Figure 7 presents the interaction between the *Aprendiz*, *Mestre* and *Sonda* agents during the reflection message production process.

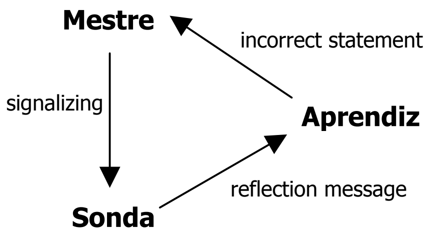

FIGURE 7 – Production of reflection message

These messages can be classified into four distinct levels, according to their objective:

- use of elements (points, segments,...) which have not been previously constructed, i.e., that do not exist. (Example message: *"What is the BC segment?"*)
- incorrect use of variables, for instance the construction of a segment using three points or the construction of a triangle with only one point. (Example message: *"How many vertices determine a triangle? Think for a while..."*)
- to write out correct statements before their derivations. (Example message: *"You cannot conclude that yet"*)
- any other situation not foreseen in the previous situations. (Example message: *"Incorrect statement"*)

6 Interactions Example

In the example that follows, we present how messages are used in the LEEG system by illustrating a possible interaction between the *Aprendiz*, *Oráculo* and *Sonda* agents during a proof of Proposition 1. One of the possible correct constructions for this proof is illustrated in Table 1 and is taken as a basis for the interactions presented below. The statements and the deduction rules written in boldface represent error situations that render the messages sent by the *Oráculo* and the *Sonda* agents.

Statement	Deduction Rule
Segment AB	
Circle A, AB	**Postulate 3 (B, AB)**

Observe that the second statement was correctly inputted, but the deduction rule is applied to an incorrect element (the center of the circle should be A and not B, as it was written out). In this case the *Mestre* agent identifies the error and signalize the *Oráculo* so that it sends the following message: *"Apply postulate 3 to point A and segment AB"*.

Circle A, AB	Postulate 3 (A, AB)
Circle B, AB	Postulate 3 (B, AB)

Using an anticipatory message, the *Mestre* could tell the *Oráculo* to send the message *"It is necessary to determine the circle intersection point A, AB and circle B, AB"*. This message anticipates the next step of the proof, avoiding possible construction mistakes.

Point C = intersection (circle A,AB; circle B,AB)	
Segment AC	Postulate 1 (A, C)
Segment BC	Postulate 1 (B, C)
Segment AC = Segment BC	

The statement Segment AC = Segment BC was inserted out of a logical and hierarchical ordering. Then the following message is sent by the *Sonda*: *"You cannot conclude that yet"*.

Segment AC = Segment AB = Segment BC	

If the *Aprendiz* makes a mistake over the same statement, the following message is sent: *"You can check the equality of segments AB and AC"*.

Segment AC = Segment AB	Definition 15 (AC, AB)
Segment BC = Segment AB	Definition 15 (BC, AB)
Segment AC = Segment BC	**Axiom 3 (AB, AC, BC)**

In this case, the deduction rule is incorrect, even though the statement was inputted correctly. Then the following message is sent: *"The deduction rule and the statement are not coherent."*

Segment AC = Segment BC	Axiom 1 (AB, AC, BC)

Again, an anticipatory message may be sent: *"You can now conclude the equality of the three sides of the triangle"*.

Segment AC = Segment AB = Segment BC	
Triangle ABC = equilateral	Definition 20 (AB, AC, BC)

When the proof is finished, the following message is sent: *"Proof concluded with success!"*.

Obviously, not all possible messages were presented in this example. We just illustrated how the system works during the knowledge construction process.

7 Concluding Remarks

The development of the work presented here is part of a conception of a proof learning system for Euclidean Plane Geometry which, as a first instance, was not thought of as an anticipatory system. However, its formalization led us to notice that we could consider it as such.

The anticipatory behavior can be observed in the *Mestre-Oráculo* and *Mestre-Sonda* interactions which follows step by step the *Aprendiz* reasoning and send messages aiming at avoiding or correcting errors.

In the LEEG system the *Aprendiz* agent was prototyped as a human entity. However, its structure allows one to study its structure as an artificial entity which should learn the logically ordered path of geometrical proofs.

Even though the system was conceived to proof learning in Geometry, its structure supports other application domains if appropriate adaptations are taken into consideration.

References

[1] Dubois D. (1999). *Review of Incursive, Hyperincursive and Antecipatory Systems – Foundation of Antecipation in Electromagnetism*. Proceedings of Computing Antecipatory Systems, CASYS'99, AIP Conference Proceedings 517, American Institute of Phisics, p.3-30.

[2] Machado J.P., De Morais C.T. & Menezes P.B. (2000). *Finite Automata: a formalism for web courses*. 13 Brazilian Symposium on Software Engineering, Florianópolis, Brasil, p.213-223.

[3] Machado J.P., De Morais C.T., Menezes P.B. & Reis, R. (2000). *Structuring web course pages as automata: revising concepts*. Recherche D'Informations Assistee par Ordinateur, Paris, C.I.D., C.A.S.I.S., v.1, p.150-159.

[4] Reitz, P. (1992). *Contribution a l'étude des environnements d'apprentissage: Conceptualisation, Spécifications et Prototipage*. Thèse de Doctorat, Université des Sciences et Techniques du Languedoc Montpellier II, France.

[5] Machado J.P., Notare M.R., Costa S., Diverio T. & Menezes P.B. (2001). *Hyper-Automaton System Applied to Geometry Demonstration Environment*. Eigth

International Conference on Computer Aided Systems Theory and Technology, Las Palmas de Gran Canaria, Spain, p.136-139 (to appear in LNCS volume).
[6] Meyer B. (1974). *An Introduction to Axiomatic Systems*. Weber & Schmidt, Boston.
[7] Costa E., Lopes M., Ferneda E. (1995). *Mathema: A Learning Environment based an a Multi-Agent Architecture*. 12 Brazilian Symposium on Artificial Intelligence, Advances in Artificial Intelligence, p.141-149.
[8] Ferneda E. (1992). *Conception d'un Agent rationnel et examen de son raisonnement en géométrie*. Thèse de Doctorat, Université des Sciences et Techniques du Languedoc Montpellier II, France.
[9] Luengo V. (1996). *Um Micromundo para la Resolución de Problemas de Demonstración en Geometría*. Conferencia Latino-Americana de Informática, Colômbia, p.941-952.
[10] Euclid. (1978). *The Thirteen Books of Euclid's Elements*. (Translation by Heath T.T.) Dover Publications, New York.
[11] Notare M. & Diverio T. (2001). *Aprendiz's Learning in Geometry Demonstration*. 7 World Conference on Computer in Education, Copenhagen, Denmark. (to appear)

Bibliografia

[ARS 96] ARSAC, Gilbert. **Un cadre d'étude du raisonnement mathématique**. 1996. Disponível em: <http://www.cabri.net/Preuve/Resumes/Arsac/Arsac96.html>. Acesso em 22 fev.2000.

[ART 88] **ARTIFICIAL INTELLIGENCE**. Un International Journal. Special Volume on Geometric Reasoning. Amsterdam, v.37, n.1-3, Dez. 1988.

[AUT 96] AUTOMATED DEDUCTION IN GEOMETRY, International Workshop, 1996, Toulouse. Selected Papers... Berlin: Springer-Verlag, 1997, 234p.

[BAL 87] BALACHEFF, Nicolas. Processus de preuve et situation de validation. **Educational Studies in Mathematics** v.18, p.147-176, 1987.

[BAL 94] BALACHEFF, Nicolas. **Didactique et Intelligence Artificialle**. 1994. Disponível por WWW em http://correio.cc.fc.ul.pt/~ulfpcost/autor/balachef.htm. Acesso em 22 fev.2000.

[BAR 95] BARBOSA, João L.M. **Geometria Euclidiana Plana**. Rio de Janeiro: Sociedade Brasileira de Matemática, 1995.

[BET 55] BETH, E.W. La Théorie de la Démonstration. In: **Les Fondements Logiques des Mathématics**. Paris: Gauthier-Villars, 1955, chap.2, p.40-64.

[BOT 61] BOTTS, Truman; PIKAART, Leonard. Mathematics from the modern viewpoint. **The Mathematics Teacher**, United States, p.498-504, Nov. 1961.

[BRA 9?] BRAZIER, F.; KEPLICZ, B.; TREUR, J.; VERBRUGGE, R**. Beliefs, Intentions and DESIRE**. 199?. Disponível em: <http://ksi.cspc.ucalgary.ca/KAW/KAW96/brazier/default.html>. Acesso em 28 fev.2000.

[CAB 99] CABRI-GÉOMÈTRE. 1999. Disponível em: <http://www-cabri.imag.fr/a-propos/index.html>. Acesso em 20 fev.2000.

[CAN 98] CANTÚ, Marco. **Dominando o Delphi 4 - A Bíblia**. São Paulo: MAKRON Books, 1998, 967p.

[CHA 9?] CHAIBEN, Hamilton. **Inteligência Artificial na Educação**. 199?. Disponível em: <http://www.cce.ufpr.br/ ~hamilton/iaed/iaed.htm>. Acesso em 25 fev.2000.

[CHO 96] CHOU, Shang-Chin; GAO, Xiao-Shan; ZHANG, Jing-Zhong. Na Introduction to Geometry Expert. In: INTERNATIONAL CONFERENCE ON AUTOMATED DEDUCTION, 13, 1996, New Brunswick. **Proceedings...**New Brunswick: Automated Deduction - CADE-13, 1996, p.235-239.

[COS 95] COSTA, Evandro; LOPES, Manoel; FERNEDA, Edilson. Mathema: A Learning Environment based an a Multi-Agent Architecture. In: BRAZILIAN SYMPOSIUM ON ARTIFICIAL INTELLIGENCE, 12, 1995, Campinas. **Proceedings...**Campinas: Advances in Artificial Intelligence, 1995, p.141-149.

[COS 98] COSTA, Evandro; PERKUSICH, Angelo; FERNEDA, Edilson. From a Tridimensional View of Domain Knowledge to Multi-agent Tutoring System. In: BRAZILIAN SYMPOSIUM ON ARTIFICIAL INTELLIGENCE, 14, 1998, Porto Alegre. **Proceedings...**Porto Alegre: Advances in Artificial Intelligence, 1998, p.61-72.

[CYR 88] CYRLUK, David; HARRIS, Richard; KAPUR, Deepak. GEOMETER: A Theorem Prover for Algebraic Geometry. In: INTERNATIONAL CONFERENCE ON AUTOMATED DEDUCTION, 9, 1988, Argone. **Proceedings...**Berlin: Springer-Verlag, 1988, p.770-771.

[DUV 91] DUVAL, R. Structure du raisonnement dédutif et apprendissage de la démonstration. **Educational Studies in Mathematics**, v.22, p.233-261, 1991.

[EUC 78] EUCLID. **The Thirteen Books of Euclid's Elements**. Traduzido por Thomas Thomas Heath. New York: Dover Publications, 1978.

[FER 92] FERNEDA, Edilson. **Conception d'un Agent rationnel et examen de son raisonnement en géométrie**.Université des Sciences et Techniques du Languedoc: Montpellier II, France, 1992. (Thèse de Doctorat).

[FRA 96] FRANKLIN, Stan; GRAESSER, Art. Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. In: THIRD INTERNATIONAL WORKSHOP ON AGENT THEORIES, ARCHITECTURES AND LANGUAGES. **Proceedings...** Berlin: Springer-Verlag, 1996.

[HER 94] HERSHkowiks, R. **Aspectos Psicológicos da Aprendizagem da Geometria**. Boletim GEPEM (Grupo de Estudos e Pesquisas em educação Matemática), 1994, nº 32, p.3-31.

[HOP 79] HOPCROFT, John; ULLMAN, Jeffrey**. Introduction to Automata Theory**. Addison-Wesley, 1979.

[JOY 98] JOYCE, D.E. **Euclid's Elements**. 1998. Disponível em: <http://aleph0.clarku.edu/~djoyce/java/elements/elements.html>. Acesso em 11 abr.2000.

[KNA 98] KNAPIK, Michael; JOHNSON, Jay. **Developing Intelligent Agents for Distributed Systems: exploring architecture, technologies and applications**. New York: McGraw-Hill, 1998.

[LAD 57] LADRIÈRE, Jean. Introduction - Les Systèmes Formel. In: **Les Limitations des Formalismes**. Paris: Gauthier-Villars Éditeur, 1957, chap.1, p13-45.

[LAN 91] LANCON, Donald. **An Introduction to the Works of Euclid with an Emphasis on the Elements**. 1991. Disponível em: <http://www.obkb.com/dcljr/euclid.html>. Acesso em 11 abr.2000.

[LIM 00] LIMA, L. F. R.; NOTARE, M. F. DIVERIO,T.A. SiMag Um sistema Multiagente para ensino de Geometria Euclidiana. In: SALÃO DE INICIAÇÃO CIENTÍFICA (12.: 2000: Porto Alegre, BR RS). **Livro de Resumos...** Porto Alegre: Pró-Reitoria de Pesquisa da UFRGS, 2000. p. 42, resumo 115.

[LUE 96] LUENGO, Vanda. Um Micromundo para la Resolución de Problemas de Demonstración en Geometría. In: CONFERENCIA LATINOAMERICANA DE INFORMÁTICA. **Proceedings...** Bogotá: [s.l], 1996, p.941-952.

[LUE 97] LUENGO, Vanda. **CABRI-EUCLIDE: Un Micromonde de Preuve intégrant la Réfutacion. Principes Didactiques et Informatiques. Réalisation**. Université Joseph Fourier: Grenoble I, França, 1997. (Thése de Doctorat).

[MAC 99] MACHADO, Júlio Henrique Araújo Pereira, PENCZEK, Leonardo, MORAIS, Carlos Tadeu Q de, MENEZES, Paulo Blauth. Autômatos finitos: um formalismo para cursos na web. In: XIII SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE, 1999, Florianópolis. XIII Simpósio Brasileiro de Engenharia de Software. **Anais...** Florianópolis: UFSC, Instituto de Informática da UFRGS, 1999. p.213-223.

[MAC 00] MACHADO, Júlio Henrique Araújo Pereira. **Hyper-Automaton: Hipertextos e Cursos na Web Utilizando Autômatos Finitos com Saída**. Porto Alegre: CPGCC da UFRGS. 2000. (Dissertação de Mestrado).

[MAC 00a] MACHADO, Júlio Henrique Araújo Pereira, MORAIS, Carlos Tadeu Q de, MENEZES, Paulo Blauth, REIS, Ricardo A L. **Structuring web course pages as automata: revising concepts**. In: RECHERCHE D'INFORMATIONS ASSISTEE PAR ORDINATEUR, 2000, Paris. Riao2000, Content-based Multimedia Information Access. Paris: C.I.D., C.A.S.I.S., 2000. v.1. p.150-159.

[MAC 01] MACHADO, Júlio; NOTARE, Márcia; DIVERIO, Tiarajú; BLAUTH, Paulo. Hyper-Automaton System Applied to Geometry Demonstration Environment. In: INTERNATIONAL CONFERENCE ON COMPUTER AIDED SYSTEMS THEORY AND TECHNOLOGY, 8, 2001, Las Palmas de Gran Canaria, Spain. **Proceedings...** Las Palmas de Gran Canaria: R. Moreno Díaz, 2001, 301p. p.136-139.

[MAC 01a] MACHADO, J. P.; NOTARE, W.R.; COSTA,S.A.; DIVERIO, T.A.; MENEZES, P.F.B. Hyper-Automaton system applied to Geometry Demonstration Environment. **Lecture Notes**. (Proceeding of EUROCAST 2001, Formal Methods and Tools for Computer Science, held in Las Palmas de Gran Canária, 13-19, Feb., 2001.)

[MCP 94] McPHEE, Nicholas; CHOU, Shang-Ching; GAO, Xiao-Shan. Mechanically proving geometry theorems using a combination of Wu's method and Collins' method. In: INTERNATIONAL CONFERENCE ON AUTOMATED DEDUCTION, 12, 1994, Nancy. **Proceedings...**Nancy: Automated Deduction - CADE-12, 1994, p.401-415.

[MEN 00] MENEZES, Paulo Blauth. **Linguagens Formais e Autômatos**. Porto Alegre: Sagra Luzzatto, 2000.

[MER 69] MESERVE, Bruce; IZZO, Joseph. **Fundamentals of Geometry**. Addison-Wesley Publishing Company, Massachusets, 1969.

[MEY 74] MEYER, Burnett. What is Mathematics?. In: **An Introduction to Axiomatic Systems**. Boston: Prindle, Weber & Schmidt, 1974, chap.1, p.1-2.

[MEY 74a] MEYER, Burnett. A brief history of the Axiomatic Method. In: **An Introduction to Axiomatic Systems**. Boston: Prindle, Weber & Schmidt, 1974, chap.2, p.3-9.

[MEY 74b] MEYER, Burnett. Axiomatic Systems. In: **An Introduction to Axiomatic Systems**. Boston: Prindle, Weber & Schmidt, 1974, chap.3, p.11-16.

[MEY 74c] MEYER, Burnett. The Affine Plane. In: **An Introduction to Axiomatic Systems**. Boston: Prindle, Weber & Schmidt, 1974, chap.4, p.17-28.

[NOT 00] NOTARE, M. R.; MENDES, S. C.; MITO, I. V.; DIVERIO, T. Uso do Computador na Educação: Um Histórico Brasileiro. In:SIMPÓSIO DE INFORMÁTICA DO PLANALTO MÉDIO, 2. Passo Fundo: 8-12, Maio, 2000. [CD-ROM] **Anais...** Passo Fundo: UPF, 2000. Disponível em CD_ROM

[NOT 00a] NOTARE, Márcia R.; DIVERIO, Tiarajú A. **Evolução Cronológica da Informática Educativa na Informática da UFRGS**. Trabalho Individual I. Porto Alegre: PPGC da UFRGS, 2000.

[NOT 00b] NOTARE, Márcia R.; DIVERIO, Tiarajú A. **Considerações sobre o Raciocínio de um Agente Racional**. Anais da V Semana Acadêmica do PPGC. Porto Alegre: PPGC da UFRGS, 2000.

[NOT 01] NOTARE, M. R.; DIVERIO, T. A. Aprendiz's learning in Geometry Demonstration. In: WORLD CONFERENCE ON COMPUTER EDUCATION, WCCE 2001, 7. **Book of Abstract**... Copenhagen, jul, 29, Aug.,3, 2001. pp.160.

[NOT 01a] NOTARE, M. R.; MACHADO, J. P.; DIVERIO, T. A.; MENEZES, P. F.B. Knowledge anticipation on agents relationship in the geometry proof system. In: INTERNATIONAL CONFERENCE ON COMPUTING ANTICIPATORY SYSTEMS, CASYS 2001, 5. **Abstract Book...** Liege, Ago., 13-18, 2002. Liege (Bélgica): HEC, 2001. Symposium 9, p.3.

[NOT 01b] Notare, M.R; Mendes, S.C.; Mito, I.V.; Diverio, T.A. Historical overview of computer usage in Brazilian Teaching. In: WORLD CONFERENCE ON COMPUTER EDUCATION, WCCE 2001, 7. **Book of Abstract...** Copenhagen, jul, 29, Aug.,3, 2001. pp.79.

[NOT 01c] Notare, M.R; Mendes, S.C.; Mito, I.V.; Diverio, T.A. Historical overview of computer usage in Brazilian Teaching. In: ANDERSON,J.; WATSON, D. **Pos Conference Book WCCE 2001**. (Held in Copenhagen, jul, 29, Aug.,3, 2001). New York: Kluver Academic Publischer, 2002. (To appear in 2002).

[NOT 01d] NOTARE, M. R.; MACHADO, J. P.; DIVERIO, T. A.; MENEZES, P. F.B. Knowledge anticipation on agents relationship in the geometry proof system. **International Journal of Computing Anticipatory Systems** (Proceeding of CASYS 2001, International Conference on Computing Anticipatory Systems, held in Liege, Belgium, Ago., 13-18, 2001.) 2002. (To appear in 2002).

[OLI 93] OLIVEIRA, Denise; HAEUSLER, Edward; PEQUENO, Tarcísio. Prova Automática de Teoremas em Dedução Natural. In: SIMPÓSIO BRASILEIRO DE INTELIGÊNCIA ARTIFICIAL, 10, 1993, Porto Alegre. **Anais...**Porto Alegre: 1993, p.111-125.

[OSI 97] OSIER, Dan; GROBMAN, Steve; BATSON, Steve. **Aprenda em 21 dias Delphi 2**. Rio de Janeiro: Campus, 1997, 840p.

[PAR 99] **PARÂMETROS CURRICULARES NACIONAIS: ENSINO MÉDIO**. Brasília, Ministério da Educação, Secretaria da Educação Média e Tecnológica, 1999, 364p.

[REI 92] REITZ, Philippe. **Contribution à l'étude des environnements d'apprentissage: Conceptualisation, Spécifications et Prototypage**. Université des Sciences et Techniques du Languedoc: Montpellier II, France, 1992. (Thèse de Doctorat).

[RUS 95] RUSSEL, Stuart; NORVING, Peter. **Artificial Intelligence – A Modern Approach**. New Jersey: Prentice-Hall, 1995.

[SAL 91] SALLANTIN, J. **Théories semi-empiriques: conceptualisation e illustrations**. Reveu d'intelligence artificielle, vol.5, nº1, p.9-67, Hermès, Paris, 1991.

[VIC 96] VICCARI, Rosa M.; GIRAFFA, Lucia M.M. Sistemas Tutores Inteligentes: abordagem tradicional x abordagem de agentes. In: SIMPÓSIO BRASILEIRO DE INTELIGÊNCIA ARTIFICIAL, 13. **Anais...**Curitiba: SBC, 1996.

[WAN 94] WANG, Dongming. Algebric factoring and Geometry Theorem Proving. In: INTERNATIONAL CONFERENCE ON AUTOMATED DEDUCTION, 12, 1994, Nancy. **Proceedings...**Nancy: Automated Deduction - CADE-12, 1994, p.386-400.

[WAN 96] WANG, Dongming. GEOTHER: A Geometry Theorem Prover. In: INTERNATIONAL CONFERENCE ON AUTOMATED DEDUCTION, 13, 1996, New Brunswick. **Proceedings...**New Brunswick: Automated Deduction - CADE-13, 1996, p.166-170.