ANALISANDO CONSTRUÇÕES NO IGEOM: UMA ABORDAGEM PARA CORREÇÃO AUTOMÁTICA DE EXERCÍCIOS

Seiji Isotani[1] *e Leônidas de Oliveira Brandão*[2]

Abstract — *The problem solving process is indispensable to learn mathematics. Nevertheless, the correction of these exercises takes a lot of time when applied with a large number of students and the manual analysis of each exercise becomes practically impossible. In this work, we presented the development of an automatic corrector of exercises in a dynamic geometry program, the iGeom, that support the authoring and the communication features. With these resources, an exercise can be created by the teacher, solved on-line by the student and the feedback is given automatically. These resources have been applied in an obligatory discipline of the degree course in Mathematics of the Institute of Mathematics and Statistics of University of São Paulo.*

Index Terms — *Dynamic Geometry, Automatic Evaluation, Authoring, Distance Education.*

INTRODUÇÃO

No processo de ensino-aprendizagem a resolução de exercícios pelos alunos e sua correção em tempo hábil por um professor é uma prática essencial para ajudá-los em seus estudos. Essa correção pode ser simplificada quando considerarmos questões do tipo multipla-escolha, preenchimento de lacunas e verdadeiro ou falso. O processo de correção e avaliação destas questões é agilizado devido à existência de corretores automáticos [1, 8, 14].

Em questões dissertativas (“abertas”) onde o aluno pode resolver o exercício de várias formas utilizando diversas estratégias, a correção manual torna-se uma tarefa trabalhosa e demorada. Da mesma forma, a correção de exercícios dissertativos de geometria (construções geométricas) não é uma tarefa simples, pois cada exercício pode ser resolvido de maneiras totalmente diferentes.

Ao considerarmos a necessidade de corrigir um grande número de exercícios dissertativos em turmas muito grandes, o esforço gasto pelo professor para avaliar, em tempo hábil, todos estes exercícios é consideravelmente alto.

Neste trabalho, apresentaremos as dificuldades encontradas na tentativa de automatizar a correção de exercícios (dissertativos) de geometria e através deste estudo mostraremos o desenvolvimento de um algoritmo que faz esta correção. Com o uso deste algoritmo no programa de geometria dinâmica iGeom, conseguimos corrigir automaticamente exercícios de geometria, tanto localmente quanto via Internet, ampliando a utilidade do iGeom e viabilizando seu acoplamento em sistemas educacionais Web que atingem um grande número de usuários.

A GEOMETRIA DINÂMICA E O IGEOM

O nome "Geometria Dinâmica" (GD) hoje é largamente utilizado para especificar a Geometria implementada em computador, a qual permite que objetos sejam movidos mantendo-se todos os vínculos estabelecidos inicialmente na construção. Este nome é melhor entendido como oposição à geometria tradicional de régua e compasso, que é "estática".

Em função desta possibilidade de alterar objetos preservando-se a construção, podemos dizer que a GD é uma geometria do tipo 1-construção, N-testes, enquanto a tradicional de régua e compasso é do tipo 1-construção, 1-teste [3]. Neste contexto, a geometria dinâmica nos oferece uma nova proposta que visa explorar os mesmos conceitos da geometria clássica, porém, através de um programa interativo [5,13].

O **iGeom: Geometria Interativa na Internet** [4], é um programa de Geometria Dinâmica que proporciona recursos facilitadores para o ensino e aprendizagem de Geometria. Além disso, o iGeom oferece ferramentas que auxiliam o professor na produção de material didático e no acompanhamento de alunos. Este programa foi desenvolvido em Java com o intuito que o mesmo pudesse ser utilizado nas formas aplicativo e *applet*.

A versão atual deste programa permite realizar todas as operações básicas de GD, como por exemplo: criar objetos geométricos (como pontos, retas, semi-retas, segmentos e circunferências); opções de edição (esconder/mostrar, remover/rastrear objetos); e opções de gravação/recuperação de arquivos em diferentes formatos (incluindo imagens em PostScript e GIF). Um recurso implementado no iGeom e que é raro nos outros programas de GD é a possibilidade de fazer macro/script com recorrência.

Além das características anteriormente apresentadas, existem outros recursos que são incorporados no iGeom para simplificar as tarefas do professor e ampliar as formas de interação com o aluno. Uma delas é a correção automática de exercícios e outras são os recursos de autoria e comunicação. Com estes recursos é possível acoplar o iGeom em sistemas Web onde o aluno pode entrar a qualquer momento para estudar, resolver exercícios e verificar seu desempenho.

[1] Seiji Isotani, Institute of Mathematics and Statistics of University of São Paulo, R. do Matão, 1010, 05508-090, São Paulo, SP, Brazil, isotani@ime.usp.br
[2] Leônidas de Oliveira Brandão,, Institute of Mathematics and Statistics of University of São Paulo, R. do Matão, 1010, 05508-090, São Paulo, SP, Brazil, leo@ime.usp.br

O PROBLEMA DA CORREÇÃO

Na aprendizagem de matemática, em geral, fazer exercícios é indispensável para um bom aprendizado. Segundo o matemático Georgo Pólya [11]: "*A Matemática é a arte de resolver problemas ... e para resolver problemas é preciso resolver problemas*". Entretanto, em turmas muito grandes, esta tarefa indispensável requer um longo período de tempo do professor que será o responsável por fazer a correção destes exercícios. Neste contexto, sem o auxílio de um sistema que automatize esta tarefa ou a existência de uma equipe de monitores, é inviável para o professor cobrar (e corrigir) muitos exercícios. Essa situação é claramente observada na disciplina "Cálculo Numérico", oferecida para aproximadamente mil alunos dos cursos de Engenharia da Escola Politécnica da Universidade de São Paulo [2].

Para ilustrar a dificuldade de se corrigir um exercício de geometria, considere o problema 1:

Problema 1. Dados dois pontos A e B, construir o ponto médio entre eles.

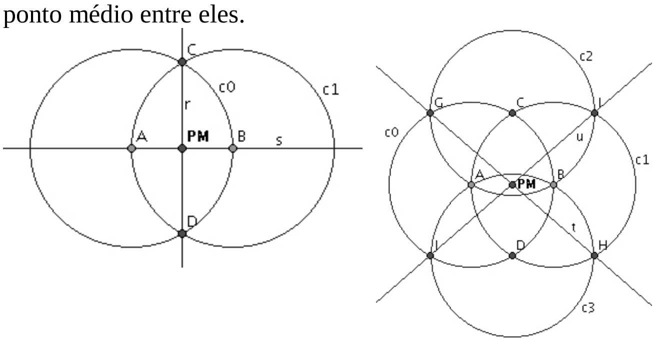

FIGURA. 1

DUAS SOLUÇÕES POSSÍVEIS PARA O PROBLEMA DO PONTO MÉDIO.

Na figura 1 são apresentadas duas construções diferentes que resolvem o problema 1. Além destas, uma infinidade de outras construções são possíveis, sendo elas minimais [3] ou não. Essa característica em exercícios de geometria dificulta ainda mais a correção automática.

Nosso objetivo ao fazer a correção automática é:

(a) Detectar como corretas, soluções que utilizam diferentes técnicas, mesmo aquelas que o professor não tenha cogitado; (b) O resultado da correção deve ser imediato; (c) Se a solução é incorreta, mostrar um contra-exemplo.

A forma de correção que utilizamos é comparar se a solução dada é "equivalente" a uma solução gabarito. Na seção "Correção Numérica" apresentamos a definição desta "equivalência" e a idéia do algoritmo implementado.

Vale destacar que não estamos querendo provar que a solução é correta, mas apenas mostrar que a solução avaliada equivale a uma outra solução previamente conhecida e, supostamente, correta (gabarito) para um problema. Para provar que uma solução é correta teríamos utilizar técnicas de *Inteligência artificial* conhecidas como *Automated Reasoning* e *Theorem Proving*, como por exemplo, as técnicas utilizadas em [6, 7, 10] que apresentam métodos para provar teoremas em geometria.

CORREÇÃO NUMÉRICA

Para se fazer a correção numérica é necessário definir o critério de **distância** entre pares de objetos geométricos de uma mesma família [4], a partir de sua **descrição computacional**. A distância pode ser representada como uma função *dist* que recebe um par de objetos geométricos $(og_1, og_2) \in F_{og}$ e retorna um valor em $\Re_+$ (equação 1).

$$\text{dist} : (og_1, og_2) \rightarrow \Re_+ \qquad (1)$$

A descrição computacional dos objetos, na maioria dos casos, é simples. Por exemplo: um ponto pode ser representado por um par (x, y), onde x e y são as coordenadas do ponto; uma reta (r(x) = ax + b), pode ser representa pelos seus coeficientes a e b; e uma circunferência pode ser representada pelas coordenadas de seu centro e a medida do raio. Deste modo, representando as famílias dos pontos, retas e circunferências, respectivamente por, $F_{og}^P, F_{og}^R, F_{og}^C$, definimos *dist* conforme a equação 2.

A partir da distância entre pares de objetos, podemos definir a distância entre pares de construções distintas, OG_p e OG_a, representadas por seus objetos, $(og_1^p, og_2^p, ..., og_i^p)$ e $(og_1^a, og_2^a, ..., og_i^a)$, conforme a equação 3.

$$\text{dist}(OG_p, OG_a) = \left\{ \sum_{i=1}^{n} \text{dist}(og_i^p - og_i^a) \right\} \qquad (3)$$

Uma solução [5] (construção geométrica) pode ser representada como uma função (equação 4) que recebe uma lista de objetos geométricos (entrada) e retorna uma outra lista de objetos geométricos (saída).

Uma vez determinada a distância entre pares de listas de objetos e a representação de uma solução, podemos definir a noção de equivalência entre duas soluções como mostra a

$$S : OG_i - OG_f \qquad (4)$$

definição 1.

Definição 1 (Equivalência). Sejam S_p e S_a duas construções sobre a mesma lista OG de objetos geométricos. Então S_a será considerada equivalente à S_p se, e somente se, para qualquer configuração OG_0 da lista OG, tivermos $\text{dist}(S_p(OG_0), S_a(OG_0)) = 0$.

[3] 3 construção minimal: qualquer construção na qual nenhum objeto pode ser removido sem comprometer o resultado.

[4] Estas famílias podem ser pontos, retas, circunferências, dentre outras.

[5] Para simplificar a notação denotaremos a aplicação de uma solução para uma lista de objetos geométricos qualquer por *S* e não S(OG)

$$dist(og_1, og_2) = \begin{cases} |x_{og1} - x_{og2}| + |y_{og1} - y_{og2}| & se\ (og_1, og_2) \in F_{og}^P x F_{og}^P \\ |a_{og1} - a_{og2}| + |b_{og1} - b_{og2}| & se\ (og_1, og_2) \in F_{og}^R x F_{og}^R \\ |x_{og1} - x_{og2}| + |y_{og1} - y_{og2}| + |r_{og1} - r_{og2}| & se\ (og_1, og_2) \in F_{og}^C x F_{og}^C \end{cases} \quad (2)$$

Utilizando as definições desta seção, dizemos que **correção numérica** é o teste de equivalência entre duas soluções. Caso ambas sejam equivalentes então a correção retorna correta, caso contrário retorna como incorreta.

O ALGORITMO

Baseado na definição de correção numérica apresentada na seção "Correção Numérica", desenvolvemos um algoritmo capaz de verificar a corretude de uma solução através de uma análise numérica dos objetos selecionados como resposta. Este algoritmo possui quatro passos principais: instanciação, transformação numérica, avaliação e validação.

As listas de objetos-resposta do professor e do aluno serão os dados de entrada do corretor e o resultado será o natural 1 (correto) ou 2 (incorreto). A execução deste algoritmo será apresentada nas duas subseções seguintes.

Transformação e Avaliação

A transformação numérica recebe um objeto geométrico e retorna uma lista de escalares que representa este objeto. Na maioria dos casos esta transformação é simples como apresentada na seção "Correção Numérica", porém alguns objetos, como o polígono, necessitam além da transformação numérica, a ordenação dos pontos que os representam. Com esta lista de escalares podemos fazer a comparação entre objetos. Assim, para analisar duas construções, o gabarito S_p do professor e a solução S_a do aluno, transformamos os objetos marcados como resposta em listas de escalares para então fazer a avaliação (figura 2).

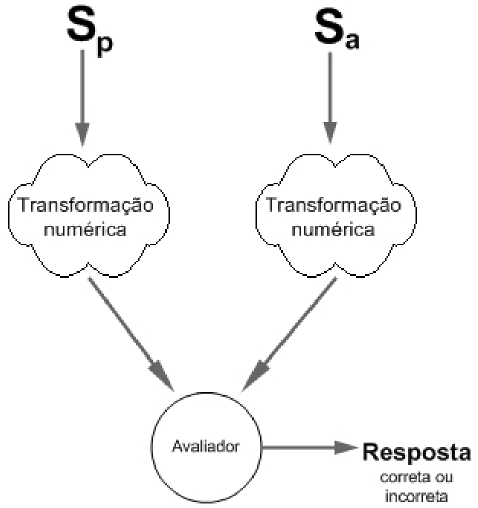

FIGURA. 2
CORREÇÃO – TRANSFORMAÇÃO NUMÉRICA E AVALIAÇÃO

A avaliação é feita em duas etapas. A primeira etapa consiste no mapeamento entre as listas e a segunda na comparação delas. A primeira etapa permite que o aluno tenha a liberdade de fazer a marcação dos objetos-resposta em qualquer ordem. Por exemplo, caso a solução do professor seja representada pelos pontos A e B e a do aluno pelos pontos C e D, a etapa de mapeamento identificará se o ponto C equivale ao ponto A ou ao ponto B.

O mapeamento dos objetos de S_a para S_p é realizado comparando cada elemento de S_a com todos os elementos de S_p que pertençam a mesma família de objetos geométricos (F_{og}) e minimizando a distância entre eles.

Como cada objeto (og^x) é transformado em uma lista de escalares (l^x), podemos então generalizar a função *dist* (equação 2), como mostra a equação 4.

$$dist(og^p, og^a) = \begin{cases} \sum_{i=1}^{n} \| l_i^p - l_i^a \| & se\ og^p, og^a \in F_{og}^m \\ \infty & c.c. \end{cases} \quad (4)$$

Assim, um objeto de S_a será mapeado para um objeto de S_p se ambos pertencerem à mesma família de objetos geométricos e a distância entre eles for a menor possível em relação aos outros objetos de S_a

A segunda etapa da avaliação consiste na comparação entre os pares de objetos geométricos mapeados (og_i^a, og_i^p) . Nesta comparação, utilizamos uma versão relaxada da definição 1 (apresentada na seção "Correção Numérica"), para levar em consideração as imprecisões numéricas ao se empregar diferentes soluções. Assim adotamos uma margem de erro ε que permite desconsiderar os erros numéricos gerados por imprecisões.

Definição 2 (Quase Equivalência). Seja S_p uma construção para um problema P, S_a uma outra construção para este problema e OG a lista de objetos geométricos de entrada. Então S_a será considerada quase equivalente à S_p e, e somente se, para qualquer configuração OG_0 da lista OG, tivermos $dist(S_p(OG_0), S_a(OG_0)) \leq \varepsilon$

Note que nesta parte do algoritmo é tratado apenas uma **instância** do exercício, considerando uma posição fixa para cada objeto de entrada. O problema em analisar apenas uma instância, aquela fornecida inicialmente, é a possibilidade do aluno apresentar uma solução próxima a do professor (sob o critério apresentado na definição 2) que, entretanto, seja “invariante” a mudanças na coordenada dos objetos de entrada. Vejamos o exemplo no problema 2:

Problema 2: Dado dois pontos A e B, construir um triângulo eqüilátero ABC.

Para resolver este problema são apresentadas duas soluções (figura 3). A primeira representa uma construção correta para o problema. A segunda é a construção de um triângulo isósceles, que coincidentemente tem um ponto C na posição tal que os segmentos AB, BC e CA possuem as mesmas medidas. Neste caso, mesmo utilizando o critério de “quase equivalência” não é possível diferenciar as soluções.

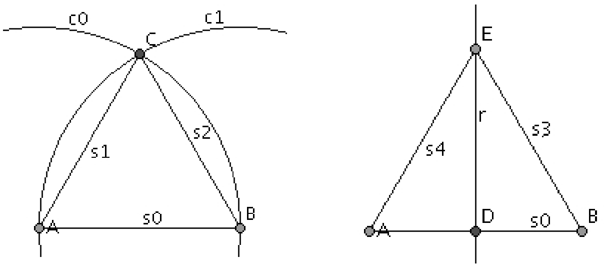

FIGURA. 3
DUAS CONSTRUÇÕES PARA O TRIÂNGULO ISÓSCELES, A PRIMEIRA ESTÁ CORRETA, ENQUANTO A SEGUNDA NÃO.

Instanciação e Validação

Uma maneira simples de detectar os erros apontados no final da seção “Transformação e Avaliação” é criar um mecanismo que analise o exercício em várias instâncias (instanciação) e somente após um número considerável de avaliações o sistema retorna o resultado da correção (validação). A simulação deste procedimento pode ser visualizada na figura 4.

A cada iteração do algoritmo de correção automática (figura 4) os objetos geométricos da entrada do exercício devem ter suas posições alteradas. Essa alteração é feita através da movimentação “aleatória” dos objetos, dentro de uma região previamente delimitada.

Ao executarmos o exercício para suas diversas instâncias, verificamos a quantidade de resultados considerados corretos e incorretos. Quando o número de resultados corretos for maior que os incorretos então retornaremos que a solução do aluno é correta, caso contrário, é considerada incorreta sendo armazenada uma instância na qual a solução foi avaliada como tal.

Esse algoritmo de correção automática foi implementado no iGeom. Outros programas de geometria dinâmica como o C.a.R [9] e o Cinderella [12] possuem algoritmos semelhantes, porém nenhum deles permite a comunicação com um servidor, o que inviabiliza o use destes programas em cursos à distância.

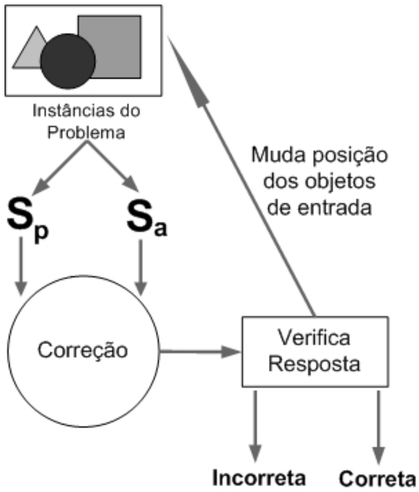

FIGURA. 4
ALGORITMO PARA CORREÇÃO AUTOMÁTICA.

AUTORIA, CORREÇÃO E COMUNICAÇÃO

O processo de criação de um exercício no iGeom possui quatro etapas: construção da solução, seleção dos objetos resposta, seleção dos objetos de entrada e salvar exercício. A interface de autoria de exercícios é bastante simples, contanto com uma janela para separar os objetos de entrada e de saída, dentre aqueles construídos na área de desenho do iGeom (figura 5). Com o exercício criado basta gravá-lo em arquivo para que qualquer usuário possa utilizá-lo ou exportá-lo para Web.

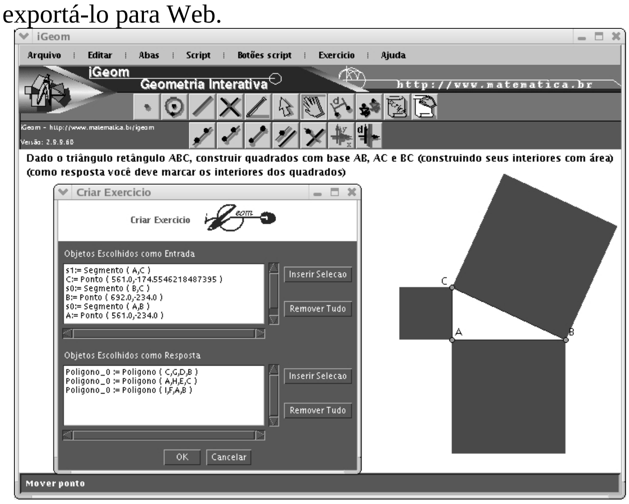

FIGURA. 5
JANELA PARA CRIAÇÃO DE EXERCÍCIOS NO IGEOM.

O iGeom utiliza o algoritmo apresentado na seção “O Algoritmo” para corrigir o exercício do aluno fazendo a comparação entre a solução do aluno e a do professor, podendo ser utilizado remotamente ou localmente.

Para que o processo de correção ocorra é necessário que o aluno pegue um arquivo gerado pelo professor e resolva-o. Quando forem selecionados os objetos-resposta, o algoritmo de correção automática será iniciado retornando: correto ou incorreto. Estes resultados poderão ser apresentados ao usuário para notificá-lo do resultado da avaliação de seu exercício, viabilizando o uso desta ferramenta de forma auto-didata, pois o aluno pode fazer e refazer exercícios, obtendo a avaliação deste processo.

Para permitir a flexibilidade e a interatividade, o programa iGeom executado como *applet* permite a comunicação com servidor. A comunicação é criada através de uma conexão HTTP [15] direta entre o *applet* e o servidor e a troca de mensagens é feita utilizando o método POST. Este recurso, em conjunto com os apresentados anteriormente, proporciona a um aluno conectado ao servidor pegar um exercício criado pelo professor, resolvê-lo e solicitar sua correção. Ao fazer esta requisição o iGeom irá corrigir o exercício (localmente) e enviar o resultado ao servidor.

Desde o início do primeiro semestre de 2004 está em teste um sistema de aprendizagem pela Web utilizando o **iGeom**, sendo utilizado por estudantes e professores em uma disciplina obrigatória oferecida para o curso de licenciatura em matemática do IME-USP, *Noções de Ensino de Matemática Usando Computador - MAC118*. Neste semestre ela foi ministrada em três turmas, uma diurna e duas noturnas, com 2 professores, 3 monitores e mais de 140 alunos.

Em edições anteriores de MAC118, todos os exercícios eram realizados utilizando o programa iGeom, mas sua correção era feita manualmente pelos monitores e professores. Devido ao número de alunos, a correção consumia grande parte do tempo dos monitores e o resultado da correção do exercício era entregue ao aluno duas ou três semanas após a realização do mesmo. Eram aplicados cerca de 20 exercícios por semestre. Com o desenvolvimento da ferramenta de correção automática no iGeom, além de reduzir o trabalho de professores e monitores, foi possível aplicar mais de 40 exercícios, desta vez, com a apresentação imediata da avaliação.

CONCLUSÕES

O propósito deste trabalho foi apresentar as dificuldades encontradas na tentativa de corrigir automaticamente exercícios de geometria, apresentar um algoritmo que consegue fazer esta correção e utilizá-lo em um sistema real de ensino de geometria. Dessa forma, conseguimos observar resultados sensivelmente positivos como, por exemplo, dobrar o número de exercícios propostos para os alunos e sua correção imediata.

Com o desenvolvimento do corretor automático no iGeom (distribuído gratuitamente através do endereço http://www.matematica.br/~igeom) e sua utilização em sistemas Web, proporcionamos um ambiente que viabiliza o oferecimento de cursos onde o número de alunos é muito grande, pois suas funcionalidades auxiliam tanto o professor, reduzindo seu trabalho no desenvolvimento do material didático, quanto o aluno que recebe o retorno da avaliação de seus exercícios quase que instantaneamente. Esta solução já está sendo aplicada em uma disciplina do curso de licenciatura em Matemática oferecida no IME-USP (http://www.ime.usp.br/~leo/mac118/04). Esta aplicação tem sido muito útil no aprimoramento do sistema, pois os alunos têm participado ativamente, sugerindo aperfeiçoamentos e detectando algumas falhas.

REFERÊNCIAS

[1] Abrão, I. C., Rayel, F., e Abrão, M. A. V. L. (2004). Questcomp: Ferramenta para avaliação de aprendizado a distância. In Proceedings of World Congress on Engineering Technology Education, http://www.inf.pucpcaldas.br/

[2] Bevilacqua, J. S. (2002). Introdução da disciplina cálculo numérico no ambiente webct. In Proceedings of VII International Conference on Engineering and Technology Education, http://www.asee.org/ international/INTERTECH2002/566.pdf.

[3] Brandão, L. O. (2004). Programação geométrica: Uso da geometria dinâmica para programação. In Anais do Segundo Colóquio de História e Tecnologia no Ensino de Matemática, pp. 191-202.

[4] Brandão, L. O. e Isotani, S.(2003). Uma ferramenta para ensino de geometria dinâmica na internet: igeom. In *Anais do Congresso da Sociedade Brasileira de Computação*, pp. 1476-1487.

[5] Braviano, G. e Rodrigues, M. H. W. L. (2002). Geometria dinâmica: uma nova geometria? Revista do Professor de Matemática, (49):22-26.

[6] Gelernter, H. (1995). Realization of a geometry-theorem proving machine. pp. 134-152.

[7] Gelernter, H., Hansen, J. R., e Loveland, D. W. (1995). Empirical explorations of the geometry-theorem proving machine. pp. 153-163.

[8] Gibson, E. J., Brewer, P. W., Dholakia, A., Vouk, M. A., e Bitzer, D. L. (1995). A comparative analysis of web-based testing and evaluation systems. In Proceedings of WWW Conference, http://renoir.csc.ncsu.edu/Faculty/Vouk/Papers/Gibson/Gibson_WWW4_1995.pdf.

[9] Grothman, R. (1999). C.A.R - Compass And Rules. http://mathsrv.kueichstaett.de/MGF/homes/ grothmann/java/zirkel/.

[10] Matsuda, N. e Vanlehn, K. (2004). Gramy: A geometry theorem prover capable of construction. Journal of Automated Reasoning, 32(1):3-33.

[11] Pólya, G. (1975). Arte de resolver problemas : um novo aspecto do método matemático. Editora Interciência, Rio de Janeiro.

[12] Richter-Gebert, J. e Kortenkamp, U. (1999). The Interactive Geometry Software Cinderella. Springer-Verlag, Heidelberg.

[13] Rodrigues, D. W. L. (2002). Uma avaliação comparativa de interfaces homem-computador em programas de geometria dinâmica. Dissertação de mestrado em engenharia de produção, Universidade Federal de Santa Catarina.

[14] Scapin, R. H. (1997). Desenvolvimento de uma ferramenta para criação e correção automática de provas na World-Wide-Web. Dissertação de mestrado em física, Instituto de Física de São Carlos, Universidade de São Paulo.

[15] W3C, W. W. W. C. (2004). HyperText Markup Language (HTML). http://www.w3.org/MarkUp/.