



DEMONSTRAÇÕES GEOMÉTRICAS AUTOMÁTICAS, SIMPLES, LEGÍVEIS E INTERESSANTES[♠]



Pedro Quaresma

CISUC, Departamento de Matemática, Universidade de Coimbra, Portugal

pedro@mat.uc.pt

[0000–0001–7728–4935]

Pierluigi Graziani

Departamento de Ciências Puras e Aplicadas, Universidade de Urbino, Itália

pierluigi.graziani@uniurb.it

[0000–0002–8828–8920]

Resumo:

A demonstração automática de teoremas é uma área de pesquisa bem estabelecida em matemática com inúmeros métodos, programas computacionais e resultados, mas também com inúmeros problemas em aberto que realçam a sua vitalidade. Entre os problemas em aberto, os três seguintes problemas estão na agenda dos especialistas na área da dedução automática: a simplicidade de uma demonstração; a legibilidade de uma demonstração; o quão interessante um dado teorema/demonstração pode ser considerado.

Palavras-chave:

Matemática; Geometria; teorema; demonstração automática.

[♠] P. Quaresma foi parcialmente suportado pela FCT – Fundação para a Ciência e Tecnologia, I.P., no âmbito do projecto CISUC – UID/CEC/00326/2020 e por Fundos Europeus, no âmbito do Programa Operacional Centro 2020. P. Graziani foi parcialmente suportado pelo Ministério da Educação de Itália, Universidade e Investigação, no âmbito do projecto PRIN 2017, “The Manifest Image and the Scientific Image”, prot. 2017ZNNW7F_004.

1. Introdução

A Matemática é um campo de conhecimento formal que tem como objecto central a demonstração matemática. De axiomas, e de regras de inferência deduzir novo conhecimento, novos teoremas, suportados por demonstrações, em que a conclusão segue logicamente do conjunto inicial de premissas.

A área da demonstração automática de teoremas (*Automated theorem proving*, ATP) é uma área de pesquisa bem estabelecida em matemática/ciências da computação, com inúmeros métodos, programas computacionais e resultados, mas também com inúmeros problemas em aberto que realçam a sua vitalidade¹. Entre os problemas em aberto, os três problemas a seguir estão na agenda dos especialistas da área da dedução automática: a simplicidade de uma demonstração; a legibilidade de uma demonstração; o quão interessante um dado teorema/demonstração pode ser considerado.

- O primeiro problema pede um critério para quantificar a simplicidade de uma demonstração.
- O segundo problema pede um critério para quantificar a legibilidade de uma demonstração.
- O terceiro problema pede um critério para quantificar o grau de interesse de um teorema e/ou uma demonstração.

A relevância destes problemas pode ser aferido pela sua menção pelo matemático David Hilbert, na sua palestra intitulada “Problemas Matemáticos”, no *2º Congresso Internacional de Matemáticos* realizado em 1900. Na versão impressa desta palestra (Hilbert, 1901; 1902), Hilbert destaca a relevância da existência de critérios gerais para a classificação de um problema matemático como um bom problema matemático, ou dito de outra forma um problema matemático interessante. Destacou também a importância da clareza e facilidade de compreensão e legibilidade, como características de um bom problema. Hilbert também enfatizou a relevância da simplicidade na matemática, sua conexão com o conceito de rigor, e o facto de o mesmo esforço de rigor nos obrigar a descobrir métodos de demonstração mais simples.

Todos esses três problemas têm seu próprio valor na matemática, em geral, no entanto, o presente texto irá abordá-los com um forte viés à área da geometria. Este texto pode ser visto como um panfleto, texto introdutório de um projecto de longo prazo, que

¹ Veja: *Conference on Automated Deduction*, <https://cadeinc.org/> e *Association for Automated Reasoning*, <https://aarinc.org/>

visa abordar estes problemas, não apenas com novas análises dos três problemas, mas também investigando as suas conexões, tanto usando estudos teóricos, como métodos empíricos, e avaliando sua implementação na demonstração e descoberta automática de teoremas em geometria.

2. Simplicidade de uma Demonstração Matemática

No texto da conferência acima citado, Hilbert analisa 23 problemas (muito mais do que os analisados durante a conferência). No entanto, Hilbert também refletiu sobre um vigésimo quarto problema dedicado à simplicidade. Esse problema foi então excluído da versão impressa, e só descoberto posteriormente em um de seus cadernos (Thiele, 2003; Thiele & Wos, 2002)²:

O vigésimo quarto problema da minha palestra em Paris seria: *Crítérios de simplicidade*, ou prova da maior simplicidade de certas demonstrações. Desenvolver uma teoria do método de demonstração em matemática, em geral. Sob um determinado conjunto de condições, só poderá existir uma demonstração considerada a mais simples. Em geral, se há duas demonstrações para um teorema, deverá ser possível derivar uma da outra, ou até que se torne bastante evidente quais são as variações (e auxiliares) a serem usadas nas duas demonstrações [...] ³ (Thiele, 2003; Thiele & Wos, 2002).

Como observou Thiele (Thiele, 2003; Thiele & Wos, 2002), Hilbert não estava sozinho em seu desejo de máxima simplicidade em demonstrações matemáticas. O matemático francês, Emile Lemoine (1840-1912), mostrou um grande interesse na simplificação das construções geométricas propondo a, *Geometrografia*, uma forma de classificar uma construção geométrica em termos de complexidade e exactidão (Lemoine 1902). Em tempos mais recentes, é possível encontrar referências ao problema da simplicidade na pesquisa de R. Thiele e L. Wos (Thiele & Wos, 2002), ou na coletânea de artigos sobre o tema editada por I. Hipólito e R. Kahle (Hipólito & Kahle, 2019).

² “Mathematisches Notizbuch”, preservado em, *Niedersaechsische Staats- und Universitaets-bibliothek Goettingen, Handschriftenabteilung* (Cod. ms. D. Hilbert 600).

³ Notebook Cod. ms. Hilbert 600:3, p. 25-26; trans. Rudiger Thiele.

2.1. Simplicidade em Geometria

Conforme observado por Thiele, E. Lemoine abordou o problema de simplicidade através daquilo que designou por *Geometrografia*.

Geometrografia

Geometrografia, “aliás a arte das construções geométricas”, visa fornecer uma ferramenta (Lemoine, 1902; Loria, 1908; Mackay, 1893; Merikoski & Tossavainen, 2010; Pinheiro, 1974; Quaresma et al., 2020; Santos, Baeta & Quaresma, 2019):

- i) para designar cada construção geométrica por um número que manifesta a sua simplicidade e exactidão⁴;
- ii) ensinar a maneira mais simples de executar uma construção;
- iii) discutir uma solução conhecida para um problema e eventualmente substituí-la por uma solução melhor;
- iv) comparar diferentes soluções para um problema, decidindo qual é a mais exacta e a solução mais simples do ponto de vista da *Geometrografia*.

Em 1888, Lemoine reduziu todas as construções geométricas executadas recorrendo a uma régua e/ou compasso, a cinco operações básicas. Considerando as modificações propostas por Mackay (Mackay, 1893), as seguintes construções de régua e compasso e os coeficientes correspondentes podem ser analisados.

- Colocar a borda da régua coincidente com um ponto R_1 .
- Colocar a borda da régua em coincidência com dois pontos $2R_1$.
- Desenhar uma linha reta R_2 .
- Colocar um ponto do compasso em um ponto determinado C_1 .
- Colocar os dois pontos do compasso em dois pontos determinados $2C_1$.
- Descrever um círculo C_2 .

Então, uma determinada construção, é medida em relação ao número de utilizações daqueles passos elementares. Para uma determinada construção tem-se:

⁴ Exactidão, ou a falta dela, neste contexto referem-se à possível imprecisão introduzida por ferramentas físicas, como uma régua ou um compasso.

$$l_1R_1 + l_2R_2 + m_1C_1 + m_2C_2$$

onde l_i e m_j são coeficientes que denotam o número de vezes que qualquer operação é executada. O número, $l_1 + l_2 + m_1 + m_2$, é designado por, *coeficiente de simplicidade (cs)* da construção, e denota o número total de operações realizadas. O número, $l_1 + m_1$, é designado por, *coeficiente de exactidão (ce)* da construção, e denota o número de operações preparatórias em que a exactidão da construção (feita com a ajuda de ferramentas físicas imprecisas) depende (Mackay, 1893; Merikoski & Tossavainen, 2010).

Geometrografia em Geometria Dinâmica

A geometrografia clássica aplica-se a construções geométricas feitas com a ajuda de uma régua e de um compasso. Sua modernização, proposta em (Quaresma et al., 2020; Santos, Baeta & Quaresma, 2019) utiliza as ferramentas dos sistemas de geometria dinâmica (DGS). Em (Quaresma et al., 2020) foi mostrado como modernizar a Geometrografia usando o *GCLC*⁵ (Janičić & Quaresma, 2006), em (Santos, Baeta & Quaresma, 2019) é mostrada a generalidade da abordagem, usando o *GeoGebra*⁶ (Hohenwarter, 2002). Considerando as operações: definir um ponto, em qualquer lugar do plano, D , e definir um determinado objecto, usando n pontos, C , temos os seguintes valores para as construções realizadas recorrendo ao *GCLC*:

- point – fixa um ponto no plano D
- line – usa dois pontos $2C$
- circle – usa dois pontos $2C$
- intersec – usa duas linhas $2C$
- intersec – usa quatro pontos $4C$
- intersec2 – usa um círculo e um círculo ou linha $2C$
- midpoint – usa dois pontos $2C$
- med – usa dois pontos $2C$
- bis – usa três pontos $3C$
- perp – usa um ponto e uma linha $2C$
- foot – usa um ponto e uma linha $2C$

⁵ <https://github.com/janicicpredrag/gclc>

⁶ <https://www.geogebra.org/>

- parallel – utiliza um ponto e uma reta 2C
- onsegment – usa dois pontos 2C
- online – usa dois pontos 2C
- oncircle – usa dois pontos 2C

Na modernização (extrapolação) da Geometrografia, considerando as “ferramentas” dos sistemas de geometria dinâmica, o coeficiente de exatidão perde seu significado. As construções serão executadas pelo DGS, portanto são precisas (exatas). Contudo, o coeficiente de simplicidade das construções ainda pode ser útil, pode ser usado para classificar construções por níveis de simplicidade. Uma nova dimensão também pode ser adicionado, o, dado pelo grau de liberdade que um determinado objeto geométrico tem, por exemplo “um ponto numa recta” tem um grau de liberdade, um ponto no plano tem dois graus de liberdade, etc. Este novo coeficiente dará um valor para o dinamismo da construção geométrica. Os graus de liberdade são medidos tendo em conta as definições dos pontos. A definição de um ponto define um ponto com dois graus da liberdade, as construções onsegment, online e oncircle, definem pontos com um grau de liberdade. Para as construções realizadas recorrendo ao *GCLC*, contidas no repositório de problemas geométricos para demonstradores automáticos de teoremas, *Thousand of Geometric problems for geometric Theorem Provers (TGTP)*⁷, obteve-se um valor médio de simplicidade, CS_{gcl} , de 20,8. Usando a análise de agrupamentos (*clustering*), k-means, função implementada no pacote estatístico do *Octave*⁸, foram definidos três agrupamentos, que descrevem um nível crescente de complexidade: construções simples, $1 \leq CS_{gcl} \leq 18$; construções de complexidade média, $18 < CS_{gcl} \leq 28$; construções complexas, $CS_{gcl} > 28$. O *TGTP* contém 71 construções simples; 81 construções de complexidade média; e 28 construções complexas.

Por exemplo, o problema (GEO0369, *TGTP*): “No triângulo ΔABC , seja F o ponto médio do lado BC , e D e E as projecções ortogonais de C e B sobre AB e AC , respectivamente. FG é perpendicular a DE em G . Mostre que G é o ponto médio de DE ”, tem uma construção geométrica com coeficiente de simplicidade 19 (ver Fig. 1), isto é, uma construção de complexidade média. O valor de 6 para seu coeficiente de liberdade é dado pelo facto de que apenas os três pontos A , B e C estão livres no plano, enquanto todos os outros pontos estão completamente vinculados, por construção.

⁷ <http://hilbert.mat.uc.pt/TGTP/index.php>

⁸ GNU Octave, versão 6.1.1, pacote, octave-statistics, função kmeans, <https://octave.sourceforge.io/statistics/function/kmeans.html>

Geometrografia na Demonstração Automática de Teoremas

A mesma abordagem pode ser (novamente) extrapolada para levar em consideração as demonstrações geométricas, ou seja, demonstrações baseadas em uma teoria geométrica axiomática, usando regras de inferência geométricas.

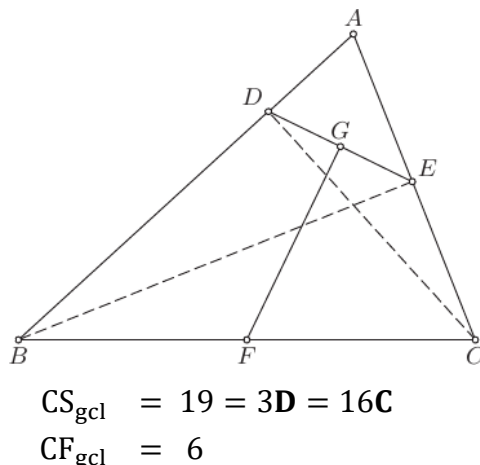


Figura 1. Coeficientes para a construção, GEO0369 (*TGTP*)

Considerando as demonstrações produzidas pelo GATP, *GCLC*, implementando o *método da área* (Janičić, Narboux & Quaresma, 2012), podemos calcular o coeficiente de simplicidade para todos os axiomas e lemas em que o método está baseado.

Além das construções geométricas nas quais as demonstrações se baseiam, (com coeficiente de simplicidade $n\mathbf{Cnst}$, existem outras etapas a serem consideradas.

- Simplificação Algébrica (Elementar) (**AS**)
- Simplificação Geométrica (Elementar) (**GS**)
- Aplicação do Lema do Método da Área n (**AML_n**)

Uma determinada demonstração pode, portanto, ser medida em relação ao número dessas etapas. Por simplificação algébrica elementar entendem-se as operações algébricas básicas: adição, subtração, multiplicação, divisão e suas propriedades de comutatividade, associatividade e distributividade. Por simplificação geométrica elementar entende-se a aplicação directa das definições das grandezas geométricas em que o método de área está baseado. Para uma dada demonstração, expressa pela equação:

$$n_1 \mathbf{Cnst} + n_2 \times \mathbf{AS} + n_3 \times \mathbf{GS} + \sum_{j=l_1}^{l_k} \mathbf{AML}_j$$

onde n_1 é o coeficiente de simplicidade da construção geométrica, n_2 é o número de simplificações algébricas e n_3 é o número de simplificações geométricas, e $\sum_{j=l_1}^{l_k} \mathbf{AML}_j$, o somatório dos coeficientes de simplicidade dos diferentes lemas usados na demonstração.

O coeficiente de simplicidade para a demonstração seria:

$$CS_{\text{proof}} = n_1 + n_2 + n_3 + \sum_{j=l_1}^{l_k} CS_{\text{proof}}(\mathbf{AML}_j)$$

O coeficiente de liberdade não tem significado neste cenário. Cada lema do método da área, \mathbf{AML}_j , tem um coeficiente de simplicidade correspondente, o termo, $\sum_{j=l_1}^{l_k} CS_{\text{proof}}(\mathbf{AML}_j)$, é a soma de todos esses valores, para todos os lemas usados na demonstração. Para conseguir isso para cada lema do método da área, calcularam-se os correspondentes coeficientes de simplicidade (Quaresma & Graziani, 2021).

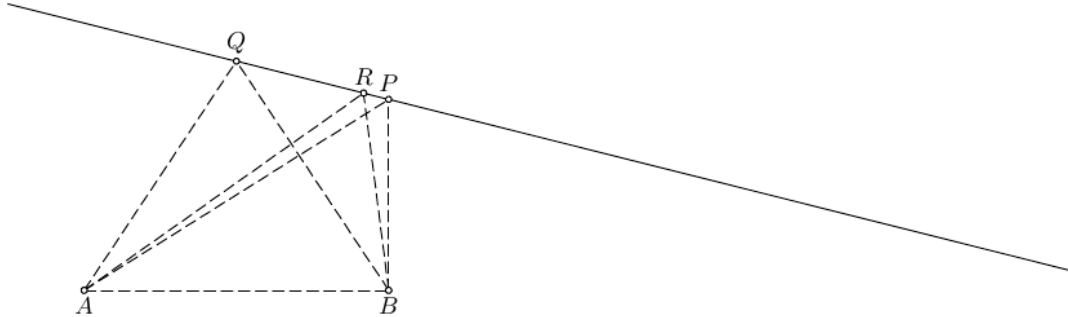
Por exemplo, a demonstração do Lema \mathbf{AML}_9 terá o seguinte coeficiente de simplicidade, $CS_{\text{proof}}(\mathbf{AML}_9) = 74^9$.

Lema (\mathbf{AML}_9). *Seja R um ponto na reta PQ. Então, para quaisquer dois pontos A e B tem-*

$$\text{se que } \mathcal{S}_{RAB} = \frac{\overline{PR}}{\overline{PQ}} \mathcal{S}_{QAB} + \frac{\overline{RQ}}{\overline{PQ}} \mathcal{S}_{PAB}.$$

O que se segue é uma versão condensada, a versão completa pode ser consultada em (Quaresma & Graziani, 2021).

⁹ O método da área está baseado em três quantidades geométricas e suas propriedades. A saber: a *área com sinal*, para o triângulo ΔABC , \mathcal{S}_{ABC} ; a *razão entre segmentos paralelos* AB e CD , $\frac{\overline{AB}}{\overline{CD}}$; e a *diferença pitagórica*, para o triângulo ΔABC , \mathcal{P}_{ABC} .

Geometrography of Lemma 9 (AML₉)

$$CS_{\text{gcl}} = 22 = 4\mathbf{D} + 18\mathbf{C}$$

$$CF_{\text{gcl}} = 8$$

- $s = \mathcal{S}_{ABPQ}$, construção inicial;
- $1 \times \mathbf{GS}$, áreas de triângulos com a mesma orientação, $\mathcal{S}_{RAB} = s - \mathcal{S}_{ARQ} - \mathcal{S}_{BPR}$;
- $1 \times \mathbf{AML}_{14}$, lema 14, $\frac{\overline{PR}}{\overline{PQ}} = r$ ($CS_{\text{proof}}(\mathbf{AML}_{14}) = 8$);
- $1 \times \mathbf{AML}_5$, lema 5, $\frac{\mathcal{S}_{ARQ}}{\mathcal{S}_{APQ}} = \frac{\overline{RQ}}{\overline{PQ}}$ ($CS_{\text{proof}}(\mathbf{AML}_5) = 18$);
- $1 \times \mathbf{GS}$, segmentos com a mesma orientação, $\frac{\overline{RQ}}{\overline{PQ}} = \frac{\overline{PQ} - \overline{PR}}{\overline{PQ}}$;
- $2 \times \mathbf{AS}$, simplificações algébricas, $\frac{\overline{PQ} - \overline{PR}}{\overline{PQ}} = (1 - r)$ e $\mathcal{S}_{ARQ} = (1 - r)\mathcal{S}_{APQ}$;
- $1 \times \mathbf{AML}_5$, lema 5, $\frac{\mathcal{S}_{BPR}}{\mathcal{S}_{BPQ}} = \frac{\overline{PR}}{\overline{PQ}}$ ($CS_{\text{proof}}(\mathbf{AML}_5) = 11$);
- $2 \times \mathbf{AS}$, simplificações algébricas, $\mathcal{S}_{BPR} = r\mathcal{S}_{BPQ}$ e $\mathcal{S}_{RAB} = s - (1 - r)\mathcal{S}_{APQ} - r\mathcal{S}_{BPQ}$;
- $2 \times \mathbf{GS}$, áreas de triângulos com a mesma orientação, $\mathcal{S}_{RAB} = s - (1 - r)(s - \mathcal{S}_{PAB}) - r(s - \mathcal{S}_{QAB})$;
- $7 \times \mathbf{AS}$, simplificações algébricas, $\mathcal{S}_{RAB} = s - s + rs + \mathcal{S}_{PAB} - r\mathcal{S}_{PAB} - rs + r\mathcal{S}_{QAB}$, $\mathcal{S}_{RAB} = r\mathcal{S}_{QAB} + (1 - r)\mathcal{S}_{PAB}$ e $\mathcal{S}_{RAB} = \frac{\overline{PR}}{\overline{PQ}}\mathcal{S}_{QAB} + \frac{\overline{RQ}}{\overline{PQ}}\mathcal{S}_{PAB}$;

Geometrografia para demonstração: $4\mathbf{D} + 18\mathbf{C} + 4\mathbf{GS} + 11\mathbf{AS} + 1\mathbf{AML}_{14} + 2\mathbf{AML}_5$

$$\mathbf{AML}_9 \begin{cases} \mathbf{CS}_{\text{proof}} & = 74 = 22 + 4 + 11 + 8 + (18 + 11) \\ \mathbf{CS}_{\text{gcl}} & = 22 \end{cases}$$

com $\mathbf{CS}_{\text{proof}}(\mathbf{AML}_{14}) = 8$, $\mathbf{CS}_{\text{proof}}(\mathbf{AML}_5) = 18$ (primeira aplicação) e $\mathbf{CS}_{\text{proof}}(\mathbf{AML}_5) = 11$ (em segunda aplicação).

Considera-se que, a partir da segunda aplicação de um lema, a sua demonstração é aceite, portanto, é necessária apenas a sua adaptação à nova configuração, ou seja, a correspondência de padrões da configuração do lema com uma nova configuração. Por essa razão, em qualquer segunda, terceira, etc. aplicação de um lema, apenas os valores do coeficiente da construção, \mathbf{CS}_{gcl} , são considerados.

Dado que uma demonstração matemática é uma sequência de passos, além do coeficiente de simplicidade, seria útil ter outros coeficientes: por exemplo, o total número de passos da demonstração; o valor da passagem mais difícil da demonstração; o número de passos diferentes de alta dificuldade na demonstração; o número de diferentes tipos de passos (lemas) da demonstração; um texto, em língua natural, da demonstração; e um valor numérico assim com um gráfico de linhas da demonstração, do ponto de vista da geometrografia.

Portanto, para caracterizar completamente uma demonstração sintética formal produzida por um GATP, podemos definir e considerar os seguintes coeficientes:

- $\mathbf{CS}_{\text{proof}}$, o coeficiente de simplicidade (como acima), dá o coeficiente de simplicidade para a demonstração, de um ponto de vista global;
- $\mathbf{CT}_{\text{proof}}$, o número total de etapas da demonstração;
- $\mathbf{CS}_{\text{proofmax}}$, o maior coeficiente de simplicidade dos lemas/definições aplicáveis, fornece o coeficiente de simplicidade para a etapa mais difícil da demonstração;
- $\mathbf{CD}_{\text{typeproof}}$, o número de diferentes tipos de lemas utilizados na demonstração;
- $\mathbf{CD}_{\text{highproof}}$, o número de diferentes etapas de alta dificuldade na demonstração;
- Um texto, em língua natural, da demonstração;
- O gráfico de linhas correspondente da demonstração no formato *tikz*¹⁰.

¹⁰ <https://ftp.eq.uc.pt/software/TeX/graphics/pgf/base/doc/pgfmanual.pdf>

É importante observar que para obter o coeficiente $CD_{\text{highproof}}$ (hp), os lemas do método de área implementados no GATP, *GCLC*, foram analisados e, usando a função de agrupamento k-means implementada no pacote de estatísticas do *Octave*, divididos em três categorias: dificuldade baixa ($hp < 284$), dificuldade média ($284 \leq hp < 1848$) e dificuldade alta ($hp \geq 1848$).

Usando os coeficientes definidos acima, temos os seguintes valores para a demonstração do lema **AML₉**:

$$\mathbf{AML}_9 \left\{ \begin{array}{l} CS_{\text{proof}} = 74 = 22 + 4 + 11 + 8 + (18 + 11) \\ CS_{\text{gcl}} = 22 \\ CT_{\text{proof}} = 19 \\ CS_{\text{proofmax}} = 18 \\ CD_{\text{typeproof}} = 2 \\ CD_{\text{highproof}} = 0 \end{array} \right.$$

Com esta extensão da Geometrografica às demonstrações produzidas pelo GATP, *GCLC*, temos a definição de um coeficiente de simplicidade para as demonstrações geométricas, produzidas automaticamente pelo GATP.

Como veremos a seguir alguns destes coeficientes vão ser importantes para a questão da legibilidade de uma demonstração geométrica.

3. Legibilidade de uma Demonstração Matemática

De importância em tudo semelhante ao problema da simplicidade, é o problema da legibilidade de uma demonstração, basta recordar obras como as de Chou S.C., Gao X.S. e Zhang J.Z. (Chou, Gao & Zhang, 1996a, 1996b), ou aquelas de Freek Wiedijk (Wiedijk, 2000)¹¹.

¹¹ Veja também (Johnson, 1957; Kane, 1967; Li & Zhang, 1999; Noonan, 1990; Smith, 1969; Stojanović, Pavlović & Janičić, 2011; Wang & Su, 2015, 2017; Y. Wang et al., 2017; Zou & Zhang, 2011). Para uma visão geral interessante, veja também (Jiang & Zhang, 2012).

3.1. Critérios de Legibilidade

Até onde sabemos, existem duas propostas precisas para medir a legibilidade de uma demonstração (Quaresma & Graziani, 2023). A primeira é a proposta por Shin-Chun Chou et al. (Chou, Gao & Zhang, 1994, 452), o factor *Maxt-Lems*, enquanto a segunda é a proposta por Freek Wiedijk e é conhecida como de *Factor de de Bruijn* (de Bruijn, 1994; Wiedijk, 2000).

Critério Maxt-Lems

Chou et al. (Chou, Gao & Zhang, 1994, 452) propuseram uma forma de medir o quão difícil é ler o texto de uma demonstração, obtida usando um provador automatizado de teoremas para geometria (GATP) que implemente o método da área (Janičić, Narboux & Quaresma, 2012). O critério *Maxt-Lems* (*ML*) considera o seguinte par, (*maxt, lems*), onde:

- *maxt* é o número de termos do polinómio maximal que ocorre na demonstração automática. Assim, *maxt* mede o número de cálculos necessários na demonstração;
- *lems* é o número de lemas de eliminação usados para eliminar pontos de grandezas geométricas. Em outras palavras, *lems* indica o número de passos de inferência na demonstração.

Usando esses dois elementos e analisando todas as demonstrações feitas pelo seu GATP, eles conseguiram determinar um limite indicativo de legibilidade. De acordo com Chou et al. (Chou, Gao & Zhang, 1994, 452) uma demonstração formal, que emprega o método da área, é considerada legível se uma das seguintes condições é válida:

- o número de termos do polinómio maximal da demonstração é menor ou igual a 5;
- o número de passos de inferência na demonstração for menor ou igual a 10;
- o termo máximo da demonstração é menor ou igual a 10 e o número de passos de inferência é menor ou igual a 20.

É interessante notar que, de acordo com o seu corpus¹²: 66,9% das demonstrações têm $\text{maxt} \leq 5$, 42,6% têm $\text{lems} \leq 10$ e 73,2% têm $\text{lems} \leq 20$.

Consideremos, por exemplo, o repositório *TGTP*, especificamente, o problema GEO0001, o *Teorema de Ceva*.

Teorema (Teorema de Ceva). *Seja ΔABC um triângulo e P um qualquer ponto no plano.*

Seja $D = AP \cap CB$, $E = BP \cap AC$ e $F = CP \cap AB$. Mostre que: $\frac{\overline{AF}}{\overline{FB}} \times \frac{\overline{BD}}{\overline{DC}} \times \frac{\overline{CE}}{\overline{EA}} = 1$. P não deve estar nas rectas paralelas a AC , AB e BC e passando por B , C e A respectivamente (Janičić, Narboux & Quaresma, 2012).

Com relação ao *critério ML*, considerando a demonstração feita pelo *GATP*, *GCLC* (Janičić & Quaresma, 2006), os valores são: $\text{maxt} = 1$ e $\text{lems} = 3$. Portanto, isso seria considerado uma demonstração legível.

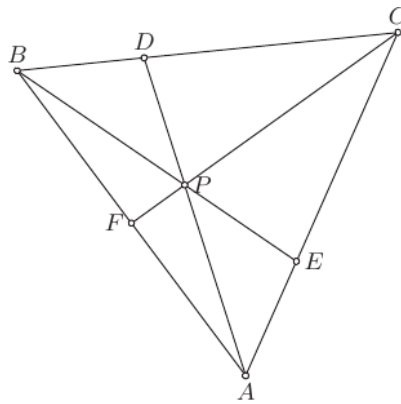


Figura 2. Teorema de Ceva, Construção Geométrica

Factor de de Bruijn

O projeto *Automath* teve como objetivo desenvolver um sistema que permitisse escrever teorias matemáticas de uma forma tão precisa que a verificação da correção dos teoremas em tais teorias poderia ser realizada por meios formais, operações

¹² Eles consideraram 478 problemas com demonstrações automáticas.

(automatizadas) aplicadas diretamente ao texto (de Bruijn, 1994). Este foi um primeiro esforço na direção da *Formalização da Matemática*, objectivo que actualmente é perseguida por pesquisadores trabalhando em sistemas como o *Coq*, *Isabelle* e *Mizar* entre outros¹³. Em “A Survey of Project Automath”, de Bruijn introduziu o *factor de perda*, diferença entre o tamanho de um texto matemático comum e sua tradução formal realizada por um programa computacional. O *factor de perda* expressa o que alguém perde, em termos de síntese (laconismo), ao traduzir matemática informal para *Automath*. Wiedijk desenvolveu o conceito e o chamou de *factor de de Bruijn*. O *factor de de Bruijn* foi desenvolvido para um situação em que uma demonstração é criada/introduzida num computador com todos os detalhes, de tal forma que o computador pode verificar sua exactidão, por exemplo, quando para um dado teorema matemático informal existente, o texto é re-escrito, “traduzido”, para uma representação formal, usando para tal um sistema como o *Automath*. Portanto, o *factor de de Bruijn* mede o quão eficiente é um tal sistema de conversão (Wiedijk, 2000). Wiedijk observou que questões não significativas sobre formatação poderiam afetar o cálculo do *factor de perda*, por exemplo: se a indentação for realizada empregando a tecla “tab”, então tal indentação pode ser oito vezes menor em comparação com situações em que a indentação é feito através da tecla de “espaço”; também o nome da macro LaTeX para o símbolo ‘ \Leftrightarrow ’ usa 15 caracteres, enquanto uma codificação como “ \Leftrightarrow ” usa apenas 3. Para que o *factor de perda* não fosse afectado por estas e outras opções de formatação, Wiedijk propôs compactar os arquivos antes de calcular a relação entre tamanhos. Wiedijk chama a relação entre textos não compactados, o *factor de de Bruijn aparente* e a relação entre textos compactados o *factor intrínseco de de Bruijn* (Wiedijk, 2000).

Consideramos que o *factor de de Bruijn* pode ser usado, num sentido mais lato, e.g. pode ser usado para medir o *factor de perda* das demonstrações matemáticas, quando produzidas por um dado GATP. Sempre que uma demonstração informal é conhecida para um determinado teorema, ela pode ser comparada com a demonstração formal produzida pelo provador automatizado de teoremas, usando uma determinada axiomatização. Isto é particularmente verdadeiro na geometria, onde uma determinada demonstração geométrica pode ser comparada com uma demonstração formal, também geométrica, produzida por um GATP.

¹³ <https://coq.inria.fr/>, <https://isabelle.in.tum.de/>, <http://mizar.org/>

Usando novamente o Teorema de Ceva como exemplo, a legibilidade da sua demonstração formal, em relação ao factor de de Bruijn pode ser calculada¹⁴ (ver Tabela 1).

	Informal	Formal	Factor de de Bruijn
Não comprimida	125KB	137KB	1,09
Comprimida	124KB	136KB	1,09

Tabela 1. Legibilidade da Demonstração do Teorema de Ceva – *GCLC*, Método da Área, *Factor de de Bruijn*

Wiedijk também introduziu o *limiar de Bruijn*, ou seja, um limite abaixo do qual “as pessoas começarão a usá-los (sistema tipo Automath) para trabalhos sérios”. Seguindo o trabalho de Wiedijk pode-se considerar o valor de 2 como um limiar de legibilidade. Mais estudos são necessários com o fim de estabelecer um limite de legibilidade para demonstrações automatizadas, usando o factor de de Bruijn, é necessário uma comparação mais ampla entre demonstrações formais e demonstrações informais. Considerando o quociente do tamanho da demonstração formal (método da área) e o tamanho da demonstração informal, o factor de de Bruijn do Teorema de Ceva é 1,09. Seria, portanto, sensato considerar a demonstração formal, legível.

Limitações do Critério ML e Factor de de Bruijn

Uma limitação surge quando se considera a seguinte classificação (Quaresma et al., 2020) das demonstrações geométricas formais produzidas pelos GATP¹⁵:

1. nenhuma demonstração legível, apenas uma saída demonstrado/não demonstrado;

¹⁴Usou-se a demonstração que pode ser encontrada em https://artofproblemsolving.com/wiki/index.php/Ceva's_theorem como fonte para a demonstração informal.

¹⁵ GATPs podem ser de dois tipos principais: algébricos, a demonstração, se existir, é feita recorrendo a um raciocínio algébrico (por exemplo, base de Gröbner); geométrica (sintética), a demonstração, se existir, é feita recorrendo a um conjunto de axiomas e regras de inferência da geometria, sem o uso de coordenadas. Métodos semi-sintéticos, por ex. o método da área, usa também os axiomas de um corpo de característica diferente de 2.

2. demonstraco no sinttica (ou seja, uma demonstraco sem uma descrio geomtrica correspondente, por ex. mtodos algbricos);
3. demonstraco semi-sinttica com traduco correspondente na linguagem do provador;
4. demonstraco (semi-)sinttica com uma correspondente representaco em linguagem natural;
5. demonstraco (semi-)sinttica com uma linguagem natural correspondente e com correspondncia visual.

Relacionando o critrio ML com esta classificaco, podemos notar que tal critrio so permite a definico de um limite para demonstraces semi-sintticas que empregam o mtodo da rea (nvel 3). A aplicabilidade direta do critrio ML a outros mtodos sintticos, por ex. mtodos de ângulo completo ou o mtodo das bases de dados dedutivas (Chou, Gao & Zhang, 1996b; Ye, Chou & Gao, 2010), seria possvel, considerando o nmero de etapas de deduco das demonstraces e adaptando a condio quanto ao nmero de termos do polinmio mximal nas demonstraces.

O factor de de Bruijn pode ser utilizado diretamente em todos os nveis acima de 1, embora  mais significativo em nveis maiores ou iguais a [sspl]. Considerando o (GCLC) e seus GATPs integrados baseados no *mtodo de rea*, *mtodo de Wu* e *mtodo das bases de Grbner* (Janii 2006),  possvel calcular a legibilidade das demonstraces desenvolvidas usando os diferentes GATPs.  possvel usar o factor de de Bruijn, para todos esses mtodos, no entanto ter-se-iam de encontrar demonstraces informais correspondentes.¹⁶

Os dois critrios analisados so muito diferentes, o primeiro  muito especfico enquanto o segundo  muito genrico.

Pe-se ento a questo: ser possvel definir um novo critrio que seja mais natural e expressivo do que o anteriores, e que possa ser generalizado para vrios mtodos de demonstraco?

A proposta abaixo baseia-se na modernizaco da *Geometrografia* (Lemoine 1902; Quaresma et al., 2020; Santos, Baeta & Quaresma, 2019).

¹⁶ O mtodo de Wu e o mtodo das bases de Grbner so ambos mtodos algbricos, do ponto de vista geomtrico as suas demonstraces so ilegveis (nvel 2).

Um Critério Geometrográfico

É interessante notar como os coeficientes geometrográficos, acima descritos (§ 2.1), destacam muitos dos aspectos salientes de uma demonstração geométrica, aspectos que podem ser usados para analisar a legibilidade de tais demonstrações. Além disso, é interessante ressaltar como o gráfico da demonstração constitui uma espécie de eletroencefalograma da máquina enquanto demonstra o teorema. Assim como um eletroencefalograma pode ser útil para medir a actividade cerebral, o gráfico da demonstração ajuda a entender o esforço dispendido na compreensão da demonstração (Quaresma & Graziani, 2023).

Aplicando a *Geometrografia* às demonstrações do método de área contidas no repositório *TGTP*, usando o *GATP GCLC*, e usando os coeficientes geometrográficos podemos argumentar em favor do seguinte coeficiente de legibilidade:

Coeficiente de Demonstrações de Legibilidade Geometrográfica (GRCP)¹⁷

$$GRCP = \left((CS_{\text{proof}} - CT_{\text{proof}}) \times (CD_{\text{highproof}} + CD_{\text{typeproof}}) \right)$$

Este coeficiente relaciona quatro quantidades: o coeficiente de simplicidade da demonstração, o número total de etapas da demonstração, o número de diferentes etapas com alta dificuldade na demonstração, o número de diferentes lemas usados na demonstração.

O primeiro factor, $(CS_{\text{proof}} - CT_{\text{proof}})$, dá uma aproximação para o coeficiente geral de simplicidade das etapas não triviais na demonstração. É possível observar que CT_{proof} conta o número de passos em vez do coeficiente de simplicidade de cada etapa. Por outro lado, em CS_{proof} , é o coeficiente de simplicidade que conta. Cada passo trivial tem um coeficiente de simplicidade igual a um, e os coeficientes de simplicidade para etapas não triviais, como a construção e os lemas, são muito maiores que um. À luz disto, pode-se concluir que a diferença entre CS_{proof} e CT_{proof} enfatiza a complexidade da demonstração, desconsiderando a sua extensão.

O segundo factor, $(CD_{\text{highproof}} + CD_{\text{typeproof}})$, dá conta do número de passos difíceis. Passos que, potencialmente, tornam a demonstração muito mais difícil de seguir, etapas onde o fluxo normal da demonstração seria interrompido para passar para a demonstração do lema, retomando após o completar da demonstração do lema. A adição

¹⁷ *Geometric Readability Coefficient of Proofs (GRCP).*

do número de etapas de alta dificuldade com o número de diferentes lemas usados na demonstração, providencia um factor multiplicador para a complexidade geral da demonstração. Uma nota final sobre este segundo factor: um passo de alta dificuldade é, com certeza, uma aplicação de um lema, no entanto sentimos que a natureza de alta dificuldade do lema em questão, é uma razão suficiente para que esta dupla contagem não seja significativa no computo geral.

Multiplicando estes factores, a aproximação para o coeficiente de simplicidade e o número de etapas difíceis — ambos os elementos que acreditamos caracterizar a legibilidade de uma demonstração — obtemos um coeficiente de legibilidade para as demonstrações geométrica produzidas por GATP.

Considerando os 71 teoremas e suas demonstrações do método de área, contidos no repositório TGTP e usando, novamente, a análise de agrupamentos, k-means, função do *Octave*, as demonstrações podem ser divididas nas seguintes classes de legibilidade geométrica:¹⁸ legível (alta legibilidade), $GRCP \leq 48000$; legibilidade média, $48000 < GRCP \leq 135000$; baixa legibilidade, $GRCP > 135000$.

O valor de $GRCP$ para o exemplo GEO0001, é: $GRCP_{GEO0001} = (220 - 32) \times (0 + 3) = 564 \leq 48000$, portanto uma demonstração legível (alta legibilidade).

Exemplo de legibilidade média do GRCP. Problema GEO0021 contido no repositório TGTP:

Teorema (Circuncentro de um Triângulo). *O circuncentro de um triângulo pode ser encontrado como a intersecção das três bissectrizes perpendiculares* tem os seguintes valores para os diferentes coeficientes.

$$GEO0021 \left\{ \begin{array}{l} CS_{proof} = 8554 \\ CS_{gcl} = 11 \\ CT_{proof} = 591 \\ CS_{proofmax} = 2807 \\ CD_{typeproof} = 13 \\ CD_{highproof} = 3 \end{array} \right.$$

$$48000 < GRCP = 127408 \leq 135000$$

¹⁸ Os valores obtidos foram arredondados.

Pelo critério *GRCP*, este é um problema de legibilidade média. Pode-se ver que ele possui 13 lemas diferentes, 3 etapas de alta dificuldade, um demonstração longa com uma diferença significativa entre o CS_{proof} e o número de etapas da demonstração (ver Figura 3).

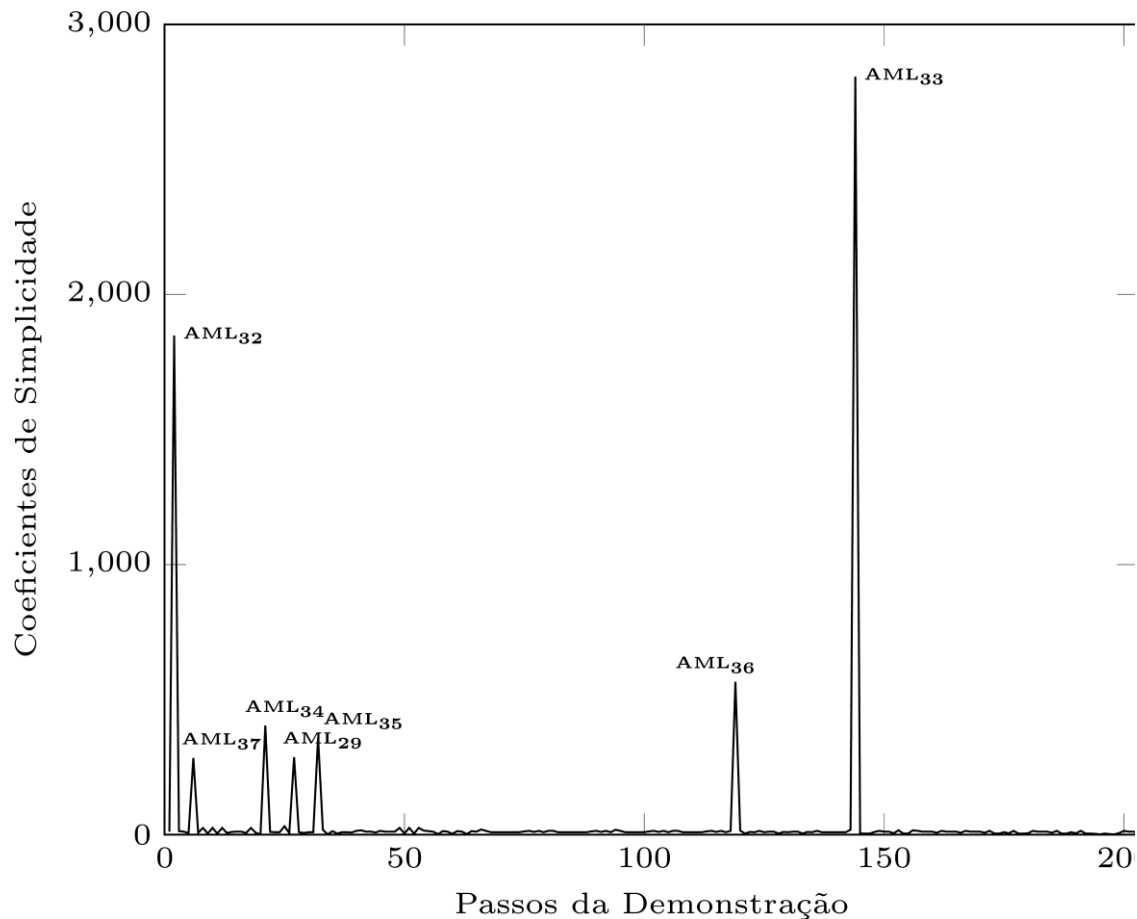


Figura 3. *TGTP, GEO0021, Circuncentro de um Triângulo*

Comparando os Diferentes Critérios

O critério do *GRCP* leva em consideração todos os aspectos significativos de uma demonstração formal, sua dificuldade geral, seu número de passos, o número de passos difíceis e o número de lemas diferentes que devem ser aplicados. Os demais critérios consideram menos aspectos. O *factor de de Bruijn*, dado seu objetivo inicial, leva em consideração apenas o tamanho da demonstração e precisa ter uma demonstração informal para comparar. O *critério ML* considera o número de diferentes lemas aplicados

e usa o número de termos do polinómio máximo como forma de se aproximar da complexidade da demonstração. Tanto no critério ML, como no critério *GRCP*, tem-se em conta o número de lemas numa demonstração: no critério *GRCP* como factor multiplicativo, no critério ML como uma das condições de legibilidade. No critério ML o número de termos no polinómio máximo são considerados, mas, como observaram os seus autores, isso mede o número de cálculos necessários na demonstração, não a sua legibilidade. A legibilidade está fracamente relacionado com o número de etapas da demonstração, o que faz é dar uma aproximação ao número de passos necessários para decompor esses longos polinómios que ocorrem na demonstração em um expressão simples.

Independentemente desta comparação dos diferentes critérios, queremos enfatizar que a visão Geométrica tem um alcance mais geral. A qualidade da abordagem Geométrica, através da análise de vários coeficientes das demonstrações, aos que se pode adicionar o gráfico da demonstração (ver Figura 3), é dada pela possibilidade de definir uma linguagem que pode ser usado por não especialistas para formular outros critérios mais fracos ou mais fortes do que aquele acima proposto. Temos a possibilidade de ir além de um dado critério geométrico, considerando uma abordagem geométrica do problema da medição da legibilidade de demonstrações formais produzidas por GATP. Uma abordagem que oferece um ambiente para analisar as demonstrações em detalhes, propor e testar critérios de legibilidade. Até onde sabemos, é o primeiro momento em que a comunidade tem acesso a uma ferramenta tão geral para o formular e o estudar da legibilidade de demonstrações formais em dedução automática em geometria. É também interessante notar que o critério *GRCP* oferece uma classificação de demonstrações que está em linha, quando considerados os pontos fundamentais, com as classificações dadas pelos outros dois critérios referidos. Isto é, as demonstrações que são classificadas como difíceis de ler de acordo ao novo critério também são classificadas como de difícil leitura para os demais, e os o mesmo se aplica a demonstrações fáceis de ler (Tabela 2).

Finalmente, temos que dizer que todos os critérios aqui propostos não têm nenhuma validação através da submissão de testes a estudantes, especialistas etc. a grande vantagem que a nossa abordagem oferece é que nos permite formular critérios que podem ser, posteriormente, implementados em repositórios como *TGTP*, podendo ser avaliado experimentalmente de uma forma muito simples.

<i>TGTP</i>	<i>ML</i>	<i>de Bruijn</i>	<i>GRCP</i>
GEO0001	$3 < 5$, passos deductivos <hr/> fácil	$1.6 < 2$ <hr/> fácil	$564 \leq 48000$ <hr/> fácil (alta)
GEO0021	$13 > 5$ passos deductivos & número de termos > 5 <hr/> difícil	$37.63 > 2$ <hr/> difícil	$127408 \leq 135000$ <hr/> difícil (média)
GEO0020	$13 > 5$ passos deductivos & número de termos > 5 <hr/> difícil	$47.31 > 2$ <hr/> difícil	$269790 > 135000$ <hr/> difícil (baixa)

Tabela 2. Comparação dos Três Critérios

4. Teoremas Matemáticos Interessantes

Como foi já referido Hilbert destacou a relevância da existência de critérios gerais para a classificação de um problema matemático como um bom problema matemático. Destacou também a importância da clareza e facilidade de compreensão, pode-se dizer, simplicidade, assim como a legibilidade, como características de um bom (interessante) problema. Larry Woss encarou, do ponto de vista computacional, esta procura da existência de critérios gerais para a classificação do quão interessante um problema matemático pode ser considerado.

Larry Woss, listou o problemas da geração automática de problemas interessantes como um dos problemas básicos na área da dedução automática (Wos, 1988).

[31º problema de Wos] Que propriedades podem ser identificadas para permitir a construção de um programa de dedução automática capaz de

encontrar teoremas novos e interessantes, em vez de simplesmente demonstrar conjecturas?

A busca de quais são as propriedades que podem ser identificadas para permitir um programa de dedução automática encontrar teoremas interessantes é um objetivo de pesquisa interessante (trocadilho intencional) (Colton, Bundy & Walsh, 2000; Gao & Cheng, 2017; Gao, Li & Cheng, 2018, 2019; Puzis, Gao & Sutcliffe, 2006; Quaresma, Graziani & Nicoletti, 2023). O 31º problema de Wos, refere os problema da geração e da classificação. No que se segue aborda-se somente a questão da classificação de um teorema como um teorema interessante, i.e. do encontrar critérios para a classificação de um problema geométrico como interessante.

A questão de gerar teoremas interessantes já foi abordado por vários autores (Colton, Bundy & Walsh, 2000; Gao & Cheng, 2017; Gao, Li & Cheng, 2018, 2019; Puzis, Gao & Sutcliffe, 2006). As diferentes abordagens encontradas na literatura (Colton, 2002; Gao, Goto & Cheng, 2014; Puzis, Gao & Sutcliffe, 2006) compartilham, em suas linhas gerais, o mesmo algoritmo: para um determinado fragmento lógico, seleciona-se um conjunto inicial de factos, referentes às propriedades intrínsecas à construção geométrica, de seguida tem-se um ciclo de geração/filtragem, até que uma dada condição de paragem seja correspondida.

Filtrando Teoremas Interessantes

Um primeiro nível de filtragem deve descartar as tautologias óbvias e também as conjecturas provadas falsas por evidências empíricas.

A filtragem por teoremas interessantes ou por teoremas desinteressantes, dois lados da mesma moeda, é feito através da aplicação de uma série de filtros. Os filtros até agora encontrados na literatura que aborda o assunto, são de natureza especulativa, não tendo ainda sido validados de nenhuma forma (Colton, Bundy & Walsh, 2000; Gao, Li & Cheng, 2018, 2019; Puzis, Gao & Sutcliffe, 2006).

Óbvio:

o número de inferências na derivação. Este filtro tenta estimar o quão óbvio é uma dada fórmula, através do número de inferências usadas na derivação da sua demonstração.

Peso:

esforço necessário para ler uma fórmula. O “peso” de uma fórmula pode ser estimado pelo número de símbolos contidos na fórmula.

Complexidade:

o esforço necessário para compreender uma fórmula. O número de símbolos de funções e predicados contidos na fórmula.

Surpresa:

mede novas relações entre conceitos e propriedades.

Intensidade:

mede o quanto uma fórmula resume as informações dos nós ancestrais em sua árvore de derivação.

Adaptabilidade:

mede quão rigorosamente as variáveis universalmente quantificadas estão restritas (para fórmulas na forma clausal).

Foco:

mede até que ponto uma fórmula gera resultados positivos ou negativos no domínio de aplicação.

Utilidade:

mede o quanto um teorema interessante contribuiu para a demonstração de outros teoremas interessantes.

Apesar da relevância destas métricas, seria apropriado construir um inquérito, com um número significativo de especialistas, com vista à sua validação. Este tipo de inquérito seria relevante não apenas para enfrentar o problema de Wos, mas também para melhor entender como construir e avaliar os programas que geram/encontram teoremas interessantes.

Resultado de Indecidibilidade

Na secção anterior foi discutida a aplicação de filtros, esses filtros são baseados em algumas medidas de interesse que ainda precisam ser validadas e que são aplicados de forma heurística. Põe-se a questão: é possível ter uma abordagem determinística, isto é, é possível escrever um programa de computador que, de forma determinística, encontre teoremas interessantes? De seguida mostramos ser não decidível determinar, para uma dada Máquina de Turing, se a linguagem por ela reconhecida tem a propriedade (não-trivial) de encontrar teoremas interessantes (Quaresma, Graziani & Nicoletti, 2023). A demonstração deste resultado usa o *Teorema de Rice* (ver Lema 1) (Rice, 1953; Rogers Jr, 1987; Sipser, 1997).

Definição 1 (Propriedade Não-Trivial). *Uma propriedade p de uma linguagem formal é não-trivial se:*

- *existe uma linguagem recursivamente enumerável com a propriedade p ;*
- *existe uma linguagem recursivamente enumerável que não possui a propriedade p .*

Lema 1 (Teorema de Rice). *Seja p qualquer propriedade não-trivial da linguagem de uma máquina de Turing. O problema de determinar se uma determinado linguagem de uma máquina de Turing tem a propriedade p é indecidível.*

Teorema 1 (Resultado de Indecidibilidade). *Para qualquer Máquina de Turing, é indecidível determinar se a linguagem por ela reconhecida tem a propriedade de encontrar teoremas interessantes.*

Demonstração. Todos os programas (máquinas de Turing) capazes de demonstração automática de teoremas em geometria e, por extensão, gerar/encontrar teoremas geométricos, dependem de uma linguagem formal para descrever as construções geométricas, conjecturas e demonstrações.

Por exemplo, podemos considerar a linguagem para a lógica de primeira ordem, *First-Order Form* (FOF)¹⁹ definida no repositório *TPTP* (Sutcliffe, 2017). Considerando de seguida as teorias axiomáticas formais para a geometria escritas nessa linguagem.

¹⁹ <http://tptp.cs.miami.edu/TPTP/QuickGuide/>

Seja p a propriedade daquela linguagem que diz que o teorema t é interessante, para qualquer definição concebível de interesse, então existe um linguagem enumerável com a propriedade p . Será suficiente restringir o linguagem de tal forma que o teorema t , e somente este, fosse reconhecido. Mas, também existe uma linguagem recursivamente enumerável que não possui a propriedade p . Bastaria restringir essa linguagem de tal forma que apenas as tautologias seriam reconhecidas. Tautologias são, para qualquer definição concebível de interesse, desinteressantes. Provamos que p , a propriedade que pode estabelecer se um determinado teorema é interessante, é uma propriedade não-trivial.

Tendo estabelecido que a propriedade p é não-trivial, então, pela aplicação de Teorema de Rice, é indecível determinar para qualquer máquina de Turing M , se a linguagem reconhecida por M tem a propriedade p .

Por outras palavras, é indecível ter um programa determinístico que possa encontrar problemas interessantes. Na melhor das hipóteses, esta é uma tarefa a ser abordada por programas baseado em algoritmos guiados por critérios heurísticos.

À luz do nosso resultado de indecibilidade, verificar se um teorema é interessante continua a ser uma tarefa inerentemente humana ou, na melhor das hipóteses, para um determinado programa de computador guiado por heurísticas. No entanto, para que este meta-nível (heurístico) possa ser definido, é preciso chegar a um grau mínimo de concordância sobre a definição de teoremas interessantes. Para alcançar algo que possa ser a base da definição de um tal meta-nível, acreditamos que uma exploração empírica da noção de interessante e do que implica concretamente é primordial. Abordaremos esta possibilidade nas conclusões.

5. Conclusões

Como já foi referido acima este é um projecto de longo prazo o qual visa abordar o problema da simplicidade e legibilidade das demonstrações geométricas assim como o problema do analisar de quão interessante um teoremas e/ou demonstração geométrica possa ser, sempre na perspectiva de um sistemas automático de demonstrações geométricas. Por esta razão, essas questões serão analisadas considerando diferentes perspectivas. A primeira etapa adotará uma visão qualitativa: utilizaremos o método da epistemologia histórica para analisar os três problemas. A epistemologia histórica é “a história do categorias que estruturam nosso pensamento, modelam nossos argumentos e demonstrações e certificam nossos padrões para a compreensão do factos. A epistemologia histórica pode (na verdade, deve ser) instanciada pela história das ideias,

mas coloca um tipo diferente de questão: não a história disto ou daquilo uso particular de, digamos, infinitesimais nas demonstrações matemáticas dos séculos XVI e séculos XVII, mas a história das mudanças nas formas e padrões da matemática manifestadas durante este período” (Daston, 1993). Usando esse método analisaremos a história das mudanças nas formas e padrões de demonstrações geométricas através de diferentes culturas matemáticas e épocas, destacando as ideias relacionadas à simplicidade e legibilidade das demonstrações geométricas, e sobre o interesse de um dado problema geométrico. Usando os dados coletados nesta primeira etapa, tentaremos definir alguns desideratos relativos aos problemas em questão. A segunda etapa adotará uma visão quantitativa: conceberemos inquéritos para serem respondidos por especialistas para melhor entender o uso dos adjetivos simples, legível e interessante, em relação às demonstrações e teoremas. Uma etapa final acabará por ser a aplicação dos resultados achados na concepção e implementação de programas que possam ir ao encontro da descoberta de novos problemas interessantes, com demonstrações simples e legíveis.

Referências

Chou, Shang-Ching, Xiao-Shan Gao & Jing-Zhong Zhang. Machine Proofs in Geometry. **World Scientific**, 1994. <https://doi.org/10.1142/2196>.

Chou, Shang-Ching, Xiao-Shan Gao & Jing-Zhong Zhang. Automated Generation of Readable Proofs with Geometric Invariants, I. Multiple and Shortest Proof Generation. **Journal of Automated Reasoning**, 17 (3): 325–47, 1996a. <https://doi.org/10.1007/BF00283133>.

Chou, Shang-Ching, Xiao-Shan Gao & Jing-Zhong Zhang. Automated Generation of Readable Proofs with Geometric Invariants, II. Theorem Proving with Full-Angles. **Journal of Automated Reasoning**, 17 (3): 349–70, 1996b. <https://doi.org/10.1007/BF00283134>.

Colton, Simon. The HR Program for Theorem Generation. In: **Automated Deduction—CADE-18**, Andrei Voronkov (Ed.), 285–89, 2002. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-45620-1_24.

Colton, Simon, Alan Bundy & Toby Walsh. On the Notion of Interestingness in Automated Mathematical Discovery. **International Journal of Human-Computer Studies**, 53 (3): 351–75, 2000. <https://doi.org/10.1006/ijhc.2000.0394>.

Daston, Lorraine. Historical Epistemology. In: **Questions of Evidence: Proof, Practice, and Persuasion Across the Disciplines**. James Chandler, Arnold I. Davidson & Harry D. Harootunian (Eds.), 282–89, 1993. The University of Chicago Press.

de Bruijn, N. G. Selected Papers on Automath. In: R. C. Vrijer, R. P. Nederpelt & J. H. Geuvers (Eds.), 133:41–161, 1994. **Studies in Logic and the Foundations of Mathematics**. Amsterdam: North-Holland.

Gao, Hongbiao & Jingde Cheng. Measuring Interestingness of Theorems in Automated Theorem Finding by Forward Reasoning: A Case Study in Peano's Arithmetic. In: **Intelligent Information and Database Systems**. Ngoc Thanh Nguyen, Satoshi Tojo, Le Minh Nguyen & Bogdan Trawinski (Eds.). 10192:115–24, 2017. Lecture Notes in Computer Science. Springer International Publishing. https://doi.org/10.1007/978-3-319-54430-4_12.

Gao, Hongbiao, Yuichi Goto & Jingde Cheng. A Systematic Methodology for Automated Theorem Finding. **Theoretical Computer Science**, 554 (October): 2–21, 2014. <https://doi.org/10.1016/j.tcs.2014.06.028>.

Gao, Hongbiao, Jianbin Li & Jingde Cheng. Measuring Interestingness of Theorems in Automated Theorem Finding by Forward Reasoning: A Case Study in Tarskis Geometry. In: **2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCOM/IOP/SCI)**. IEEE. 2018. <https://doi.org/10.1109/SmartWorld.2018.00064>.

Gao, Hongbiao, Jianbin Li & Jingde Cheng. Measuring Interestingness of Theorems in Automated Theorem Finding by Forward Reasoning Based on Strong Relevant Logic. In: **2019 IEEE International Conference on Energy Internet (ICEI)**, 356–61. IEEE. 2019. <https://doi.org/10.1109/ICEI.2019.00069>.

Hilbert, David. Mathematische Probleme. **Arch. Für Math. Phys**, 3 (1): 44–63, 213–37, 1901.

Hilbert, David. Mathematical Problems. **Bulletin of the American Mathematical Society**, 8: 437–79, 1902.

Hipólito, Inês & Reinhard Kahle (Eds.). **The notion of simple proof – Hilbert’s 24th problem**. Vol. 377, 2019. Philosophical Transactions of the Royal Society A.

Hohenwarter, M. GeoGebra - a Software System for Dynamic Geometry and Algebra in the Plane. **Master’s thesis, Austria**: University of Salzburg, 2002.

Janičić, Predrag. GCLC — A Tool for Constructive Euclidean Geometry and More Than That. In: **Mathematical Software - ICMS 2006**. Andrés Iglesias & Nobuki Takayama (Eds.), 4151:58–73, 2006. Lecture Notes in Computer Science. Springer.
https://doi.org/10.1007/11832225_6.

Janičić, Predrag, Julien Narboux & Pedro Quaresma. The Area Method: A Recapitulation. **Journal of Automated Reasoning**, 48 (4): 489–532, 2012.
<https://doi.org/10.1007/s10817-010-9209-7>.

Janičić, Predrag & Pedro Quaresma. System Description: GCLCprover + GeoThms. In: **Automated Reasoning**. Ulrich Furbach & Natarajan Shankar (Eds.), 4130:145–50, 2006. Lecture Notes in Computer Science. Springer. https://doi.org/10.1007/11814771_13.

Jiang, Jianguo & Jingzhong Zhang. A Review and Prospect of Readable Machine Proofs for Geometry Theorems. **Journal of Systems Science and Complexity**, 25 (4): 802–20, 2012. <https://doi.org/10.1007/s11424-012-2048-3>.

Johnson, Donovan A. The Readability of Mathematics Books. **The Mathematics Teacher**, 50 (2): 105–10, 1957. <https://doi.org/10.5951/MT.50.2.0105>.

Kane, Robert B. The Readability of Mathematical English. **Journal of Research in Science Teaching**, 5 (3): 296–98, 1967. <https://doi.org/10.1002/tea.3660050319>.

Lemoine, Émile. **Géométriegraphie Ou Art Des Constructions Géométriques**. Vol. 18, 1902. Phys-Mathématique. Scientia. <http://catalogue.bnf.fr/ark:/12148/cb36049032t>.

Li, Chuan-Zhong & Jhing-Zhong Zhang. Readable Machine Solving in Geometry and ICAI Software MSG. In: **Automated Deduction in Geometry**, 67–85, 1999. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-47997-x_5.

Loria, Gino. La Geometrografia e Le Sue Trasfigurazioni. **Period. Math.**, 3 (6): 114–22, 1908.

Mackay, J. S. The Geometrography of Euclid's Problems. **Proceedings of the Edinburgh Mathematical Society**, 12: 2–16, 1893. <https://doi.org/10.1017/S0013091500001565>.

Merikoski, K. Jorma & Timo Tossavainen. Two Approaches to Geometrography. **Journal for Geometry and Graphics**, 13 (1): 15–28, 2010.

Noonan, James. Readability Problems Presented by Mathematics Text. **Early Child Development and Care**, 54 (1): 57–81, 1990. <https://doi.org/10.1080/0300443900540104>.

Pinheiro, Virgílio Athayde. **Geometrografia 1**. Bahiense, 1974.

Puzis, Yury, Yi Gao & G. Sutcliffe. Automated Generation of Interesting Theorems. In: **FLAIRS Conference**, 2006.

Quaresma, Pedro & Pierluigi Graziani. The Geometrography's Simplicity Coefficient for the Axioms and Lemma of the Area Method. **Technical Report TR 2021-001**. Center for Informatics; Systems of the University of Coimbra, 2021.

Quaresma, Pedro & Pierluigi Graziani. Measuring the Readability of Geometric Proofs—the Area Method Case. **Journal of Automated Reasoning**, 67 (1), 2023. <https://doi.org/10.1007/s10817-022-09652-0>.

Quaresma, Pedro, Pierluigi Graziani & Stefano M. Nicoletti. Considerations on Approaches and Metrics in Automated Theorem Generation/Finding in Geometry. **ADG2023**. 2023.

Quaresma, Pedro, Vanda Santos, Pierluigi Graziani & Nuno Baeta. Taxonomy of Geometric Problems. **Journal of Symbolic Computation**, 97 (March): 31–55, 2020. <https://doi.org/10.1016/j.jsc.2018.12.004>.

Rice, H. G. Classes of Recursively Enumerable Sets and Their Decision Problems. **Transactions of the American Mathematical Society**, 74 (2): 358–66, 1953. <https://doi.org/10.2307/1990888>.

Rogers Jr, Hartley. **Theory of Recursive Functions and Effective Computability**. MIT press, 1987.

Santos, Vanda, Nuno Baeta & Pedro Quaresma. Geometrography in Dynamic Geometry. **International Journal for Technology in Mathematics Education**, 26 (2): 89–96, 2019. https://doi.org/10.1564/tme_v26.2.06.

Sipser, Michael. **Introduction to the Theory of Computation**. PWS Publishing Company, 1997.

Smith, Frank. The Readability of Junior High School Mathematics Textbooks. **The Mathematics Teacher**, 62 (4): 289–91, 1969. <https://www.jstor.org/stable/27958126>.

Stojanović, Sana, Vesna Pavlović & Predrag Janičić. A Coherent Logic Based Geometry Theorem Prover Capable of Producing Formal and Readable Proofs. In: **Automated Deduction in Geometry**. Pascal Schreck, Julien Narboux & Jürgen Richter-Gebert (Eds.), 6877:201–20, 2011. Lecture Notes in Computer Science. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-25070-5_12.

Sutcliffe, Geoff. The TPTP Problem Library and Associated Infrastructure. **Journal of Automated Reasoning**, 59 (4): 483–502, 2017. <https://doi.org/10.1007/s10817-017-9407-7>.

Thiele, Rüdger. Hilbert's Twenty-Fourth Problem. **The American Mathematical Monthly**, 110 (1): 1, 2003. <https://doi.org/10.2307/3072340>.

Thiele, Rüdger & Larry Wos. Hilbert's Twenty-Fourth Problem. **Journal of Automated Reasoning**, 29 (1): 67–89, 2002. <https://doi.org/10.1023/A:1020537107897>.

Wang, Ke & Zhendong Su. Automated Geometry Theorem Proving for Human-Readable Proofs. In: **Proceedings of the 24th International Conference on Artificial Intelligence**, 1193–99, 2015. IJCAI'15. AAAI Press. <http://dl.acm.org/citation.cfm?id=2832249.2832414>.

Wang, Ke & Zhendong Su. Automated Geometry Theorem Proving for Human-Readable Proofs. **ArXiv**, 2017.

Wang, Ying, Yongsheng Rao, Hao Guan & Yu Zou. NetPad: An Online DGS for Mathematics Education. In: **2017 12th International Conference on Computer Science and Education (ICCSE)**. IEEE, 2017. <https://doi.org/10.1109/ICCSE.2017.8085507>.

Wiedijk, Freek. The de Bruijn Factor. **Poster at International Conference on Theorem Proving in Higher Order Logics (TPHOL2000)**, 2000.

Wos, Larry. **Automated Reasoning: 33 Basic Research Problems**. Prentice-Hall, 1988
Ye, Zheng, Shang-Ching Chou & Xiao-Sha Gao. Visually Dynamic Presentation of Proofs in Plane Geometry, Part 2. **Journal of Automated Reasoning**, 45: 243–66, 2010. <https://doi.org/10.1007/s10817-009-9163-4>.

Zou, Yu & Jingzhong Zhang. Automated Generation of Readable Proofs for Constructive Geometry Statements with the Mass Point Method. In: **Proceedings of the 8th International Conference on Automated Deduction in Geometry**, 221–58, 2011. ADG'10. Berlin, Heidelberg: Springer-Verlag. https://doi.org/10.1007/978-3-642-25070-5_13.



