

Egy nemtriviális geometriai tétel nemtriviális számítógépes bizonyítása

KOVÁCS ZOLTÁN

Több mint 25 év telt el, mióta a Polygon hasábjain két fiatal szegedi matematikus – akkor még egyetemisták – egy korábban ismeretlen állítás bizonyítását közzölték [1]. A tétel az Euler-egyenessel volt kapcsolatos, mely a KöMaL 1993. évi 3. számában szereplő F. 2953. feladatnak volt egy általánosítása.

Néhány évvel később korábbi középiskolai tanárunk egy internetes cikkben újította meg az érdeklődést a téma iránt [2]. A két diák azóta matematikus-kutatóként a matematika más területein kamatoztatja tudását. Akkori munkájuk azonban tovább él, és az általuk bizonyított tétel egy új matematikai ágra ma is inspirálóan hat: az *automatikus bizonyítások* gyakorlati módszereire.

Az alábbiakban tételük egy koordinátageometriai alapú, polinomalgebrai levezetését mutatjuk be. A teljes levezetés majdnem 900 oldalt foglal magába, melyet egy mai átlagos PC kb. 20 másodperc alatt hozott létre teljesen automatizált módon, miután a feladatot *GeoGebra*-szerkesztésként megfogalmaztuk. A levezetés kb. 200 elemi lépésből áll, amelyből a feladatot szövegesen mintegy 40 lépésben tisztázza a gép, továbbá 100 lépés ezek matematikailag pontos megfogalmazása egyenletek segítségével. Ezután mintegy 50 lépésben egy olyan konkrét speciális esetet ad meg a program, ami nem szorítja meg az általánosságot. A fennmaradó 5 lépés – ebből az egyik lépés teszi ki az oldalak 99 %-át – egy 54 ismeretlenes 14-edfokú algebrai kifejezés elemzése, melyet egy komputeralgebrai szoftverrel leellenőrizhetünk, egyszerűsíthetünk. Mi ehhez a *Mathematicát* (illetve annak ingyenesen elérhető változatát, a *wolframscript*-et) választva az ellenőrzést kb. 30 másodperc alatt hajtottuk végre ugyanazon a számítógépen.

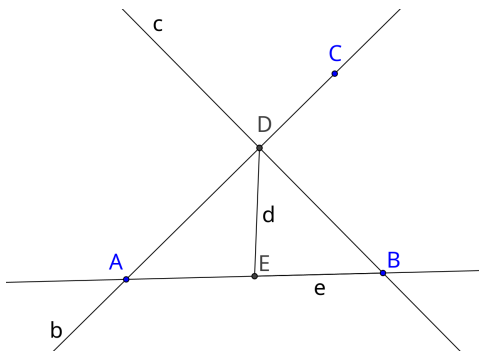
Módszerünket először egy jóval egyszerűbb példán illusztráljuk. Ez azt is mutatja, hogy egyszerűbb esetekben akár iskolai órán vagy a felsőoktatásban is bevethető ilyen típusú bizonyítás.

1. A módszer

1.1. Koordinátageometriai átírás

A legtöbb olvasó bizonyára jól ismeri a Thalész-tétel megfordítását: *Minden derékszögű háromszög körülírható körének átmérője a háromszög átfogója.* Mi most

ezt egy kissé meglepő formában fogalmazzuk meg: Tekintsünk három tetszőleges pontot a síkon: A -t, B -t és C -t, valamint egy olyan AC -re eső D pontot, amelyre $AD \perp BD$. Legyen E az AB szakasz felezőpontja, ekkor szükségképpen $BE = DE$ (l. 1. ábra, itt a GeoGebrát használtuk a szerkesztéshez).



1. ábra. A Thalész-tétel megfordításának egy lehetséges szemléltetése GeoGebrában

Ha most a GeoGebra egy kísérleti változatában, a *GeoGebra Discovery*ben¹ a *BizonyításLépései*($d == e$) paranccsal dolgozunk tovább (ezt úgy kell kiadni, hogy a *CAS - komputeralgebra* ablak aktív legyen, de magát a parancsot nem ott, hanem a *Parancssorban* vagy az *Algebra* ablakban kell beírni), akkor a következő levezetéshez jutunk²:

A Thalész-tétel megfordításának egy lehetséges levezetése

1. Legyenek A , B , C tetszőleges pontok.
2. Legyen b az A , C egyenes.
3. Legyen c a B -n keresztül b -re merőleges egyenes.
4. Legyen D a b és c metszete.
5. Legyen E az A , B középpontja.

¹A szoftver online elérhető a <https://autgeo.online> címen, vagy letölthető a <https://github.com/kovzol/geogebra/releases/> weboldalról.

²Ez a levezetés, mint ahogy a mellékletben szereplő is, teljes egészében számítógép által generált. Meghagytuk a gép által létrehozott eredeti formát, amelyben a változók indexei nincsenek alsó indexbe téve, csak közvetlenül a változók mellett jelennek meg. Ezt az írásmódot a cikk további részében nem alkalmazzuk, tehát pl. a gép által generált v_1 változót következetesen v_1 -ként írjuk a cikk többi részében, ami áttekinthetőbb megjelenést eredményez.

6. Legyen d az $[E, D]$ szakasz.
7. Legyen e az $[E, B]$ szakasz.
8. Bizonyítsuk be, hogy $d \stackrel{?}{=} e$.
9. Az állítás igaz, ha a szerkesztés nem elfajuló (lásd lejjebb).
10. Indirekt bizonyítást adunk.
11. Jelöljük az A szabad pontot $(v1, v2)$ -vel.
12. Jelöljük a B szabad pontot $(v3, v4)$ -gyel.
13. Jelöljük a C szabad pontot $(v5, v6)$ -tal.
14. A c objektumhoz a következő további változókra lesz szükség:
15. $v7$: A B -ben b -re merőlegesen állított egyenes másik pontjának x -koordinátája
16. $v8$: A B -ben b -re merőlegesen állított egyenes másik pontjának y -koordinátája
17. $e1 := -v8 + v5 + v4 - v1 = 0$
18. $e2 := -v7 - v6 + v3 + v2 = 0$
19. A $D = \text{Metszéspont}(b, c)$ definíciót figyelembe véve:
20. Jelöljük a D függő pontot $(v9, v10)$ -zel.
21. $e3 := -v9 * v6 + v10 * v5 + v9 * v2 - v5 * v2 - v10 * v1 + v6 * v1 = 0$
22. $e4 := v9 * v8 - v10 * v7 - v9 * v4 + v7 * v4 + v10 * v3 - v8 * v3 = 0$
23. Az $E = \text{Középpont}(A, B)$ definíciót figyelembe véve:
24. Jelöljük az E függő pontot $(v11, v12)$ -vel.
25. $e5 := 2 * v11 - v3 - v1 = 0$
26. $e6 := 2 * v12 - v4 - v2 = 0$
27. Következmény: $d \stackrel{?}{=} e$, algebrai formában:
28. $T1 := -2 * v12 * v10 + v10^2 - 2 * v11 * v9 + v9^2 + 2 * v12 * v4 - v4^2 + 2 * v11 * v3 - v3^2 = 0$

29. A következmény tagadása:
30. v_{13} : extra változó a tagadáshoz
31. $(T1 * v_{13} - 1) = 0$
32.
$$e7 := -1 - 2 * v_{13} * v_{12} * v_{10} + v_{13} * v_{10}^2 - 2 * v_{13} * v_{11} * v_9 + v_{13} * v_9^2 + 2 * v_{13} * v_{12} * v_4 - v_{13} * v_4^2 + 2 * v_{13} * v_{11} * v_3 - v_{13} * v_3^2 = 0$$
33. Az általánosság megszorítása nélkül néhány koordináta rögzíthető:
34. $v_1 = 0, v_2 = 0, v_3 = 0, v_4 = 1$
35. Az állítás a következő nem-elfajuló feltételek esetén lesz igaz:
36. • az ABC háromszög nem elfajuló
37. $endg := -1 - v_{14} * v_5 = 0$
38. Behelyettesítés után:
39. $sndg := -1 - v_{14} * v_5 = 0$
40. Az állítás bizonyos feltételek mellett érvényes:
41. • A és B nem egyeznek meg
42. Az előfeltételek és tagadott következmény behelyettesítés után:
43. $s1 := 1 - v_8 + v_5 = 0$
44. $s2 := -v_7 - v_6 = 0$
45. $s3 := -v_9 * v_6 + v_{10} * v_5 = 0$
46. $s4 := -v_9 + v_9 * v_8 + v_7 - v_{10} * v_7 = 0$
47. $s5 := 2 * v_{11} = 0$
48. $s6 := -1 + 2 * v_{12} = 0$
49.
$$s7 := -1 - v_{13} + 2 * v_{13} * v_{12} - 2 * v_{13} * v_{12} * v_{10} + v_{13} * v_{10}^2 - 2 * v_{13} * v_{11} * v_9 + v_{13} * v_9^2 = 0$$
50. Most tekintsük a következő egyenletet:

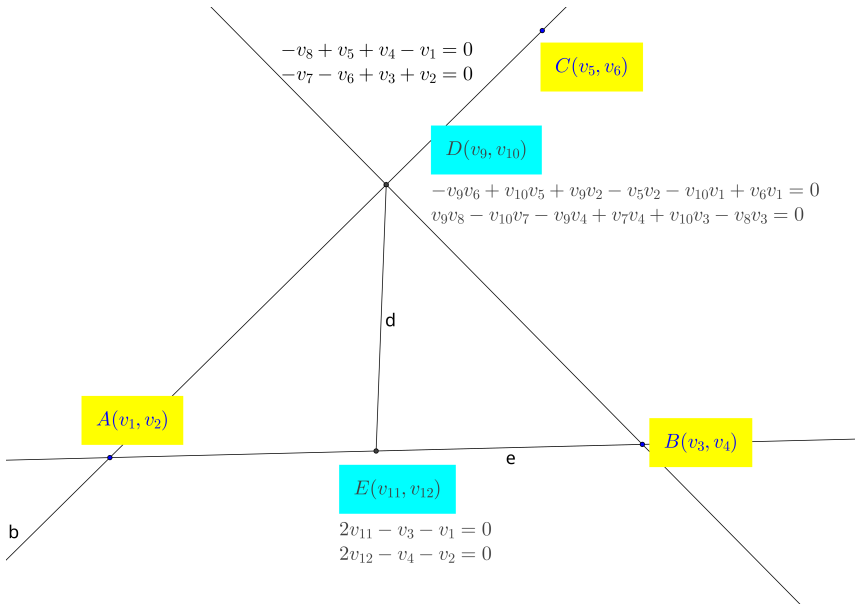
$$51. s1*(-v13*v14*v9^2)+s2*(v10*v13*v14*v9-v13*v14*v9)+s3*(-v13*v14^2*v5-v10*v13*v14)+s4*(-v13*v14*v9)+s5*(-v13*v9)+s6*(-v10*v13+v13)+s7*(-1)+s8dg*(-v10*v13*v14*v5+v13*v14*v6*v9-v13*v9^2-v10^2*v13+v10*v13)$$

$$1 = 0$$

52. **Ellentmondás! Ezáltal az eredeti állítást igazoltuk.**

53. Az állítás nehézségi foka: 4.

Látható, hogy a geometriai tétel teljesen átfogalmazódott koordinátagéometriai állítássá. Hogy az egyes változók jelentését jobban áttekinthessük, a beírt GeoGebra parancsnak adhatunk egy második paramétert is: ha oda `true`-t írunk, akkor a 2. ábrán látható segítséget is kirajzoltathatjuk.



2. ábra. Segítő ábra a Thalész-tétel megfordításának koordinátagéometriai formájában megjelenő változókhoz és egyes egyenletekhez

Erről az ábráról a (v_7, v_8) pont hiányzik (ez egy, a GeoGebra által önhatalmúlag felvett pont, ami a $c(=BD)$ egyenesre illeszkedik), de a hozzá tartozó koordiná-

tageometriai definíció az ábra felső részén látható (l. a levezetés 17–18. lépéseit is, amelyek a két egyenletet e_1 -gyel és e_2 -vel is jelzik). Mi a továbbiakban ezt a program által nem jelölt pontot X -szel jelöljük.

A D pont úgy lett megadva, hogy az $ACD\triangle$ és a $BXD\triangle$ egyaránt 0 területű legyen. Ez biztosítja, hogy D egyaránt az AC és a BX egyenesekre is illeszkedjen, vagyis hogy $D = AC \cap BX$. A használt területképletek azon olvasók számára magától értetődőek, akik járatosak a determinánsok használatában, itt a

$$\det \begin{pmatrix} v_1 & v_5 & v_9 \\ v_2 & v_6 & v_{10} \\ 1 & 1 & 1 \end{pmatrix} = 0 \quad \text{és} \quad \det \begin{pmatrix} v_3 & v_7 & v_9 \\ v_4 & v_8 & v_{10} \\ 1 & 1 & 1 \end{pmatrix} = 0$$

azonosságokról van szó, de ezek az egyenletek közvetlenül is levezethetők vektorok párhuzamosságát felhasználva. Ez a két egyenlet a levezetésben a 21–22. lépésben kerül bevezetésre e_3 és e_4 egyenletként.

Az E pont az AB felezőpontja, így koordinátái is „félúton” vannak A és B között. Az ábrán és a levezetésben is látható mindkét egyenlet, utóbbiban e_5 és e_6 -ként a 25–26. lépésben.

A vizsgálandó állítás előfeltételeit maradéktalanul felírtuk algebrai formában. A bizonyítandó következmény algebrai formája az, hogy a BE szakasz ugyanolyan hosszú, mint a DE szakasz, és ezt egy egyszerű Pitagorasz-tétellel ki is tudjuk fejezni:

$$(v_3 - v_{11})^2 + (v_4 - v_{12})^2 = (v_9 - v_{11})^2 + (v_{10} - v_{12})^2.$$

Ez azonos a levezetés 28. sorában felírt T_1 állítással.

Módszerünk egyik fontos alappillére, hogy indirekt módon bizonyítunk. Ehhez a T_1 állítást fogjuk tagadni, a tagadást viszont olyan módon kell megoldani, hogy a polinomegyenletek körében maradjunk. Miért? Mert az alkalmazott komputer-algebrai módszer csak polinomegyenletek halmazára fog működni. Szerencsére az egyenletek tagadására van bejáratott technika, melyet az irodalom több helyen csak Rabinowitsch-féle trükként [3] emleget: Egy

$$p(x_1, x_2, \dots, x_n) = 0$$

alakú polinomegyenlet egy lehetséges tagadása a

$$p(x_1, x_2, \dots, x_n) \cdot x_{n+1} - 1 = 0$$

polinomegyenlet (ahol egy új változót, x_{n+1} -et vezetünk be), vagyis a két polinomegyenlet egymás negáltja. Ennek végiggondolását az olvasóra bízunk. A levezetés 31. és 32. sorában pontosan a $T_1 = 0$ polinomegyenlet tagadása jelenik meg, melyet a program e_7 -tel jelöl.

Egy további fontos lépés, hogy az állítást elegendő lehet belátni olyan esetekre, amikor bizonyos szabad pontokat fix koordinátákkal határozzuk meg. A Thalész-tétel megfordítása tipikusan ilyen: az állítás igaz vagy hamis volta független attól,

hogy hol helyezkednek el az A , B és C pontok, vagyis van értelme annak, hogy az A pontot mondjuk a $(0,0)$ -ba, a B -t pedig a $(0,1)$ -be tegyük, a C pontot pedig továbbra is szabadon fogjuk választani. Ezeket a fix koordinátákat a levezetés 34. sorában látjuk.

1.2. Elfajuló esetek, hitelesített bizonyítás

Ha most behelyettesítjük a $v_1 = 0$, $v_2 = 0$, $v_3 = 0$ és $v_4 = 1$ értékeket az $e_1, e_2, e_3, e_4, e_5, e_6$ és e_7 egyenletekbe, új egyenleteket kapunk $s_1, s_2, s_3, s_4, s_5, s_6$ és s_7 jelölésekkel (43–49. sor). Most amennyiben találunk egy olyan

$$(1) \quad s_1 \cdot f_1 + s_2 \cdot f_2 + s_3 \cdot f_3 + s_4 \cdot f_4 + s_5 \cdot f_5 + s_6 \cdot f_6 + s_7 \cdot f_7$$

összeget, amelyben $f_1, f_2, \dots, f_7$ mind polinomok, és az összeg egy konstans érték (pl. 1), akkor nyilvánvalóan ellentmondásra jutnánk, mert (1) nyilvánvalóan 0 kéne legyen, hiszen $s_1 = s_2 = \dots = s_7 = 0$.

A konkrét példánknál még van egy pici bökkenő: ilyen $f_1, f_2, \dots, f_7$ polinomokat sajnos nem találunk, mert az előfeltételeink ebben a formában még nem ellentmondóak, ugyanis bizonyos elfajuló helyzetekben a Thalész-tétel megfordítása nem lesz igaz. Később részletezni fogjuk, hogyan tudjuk az előfeltételeinket úgy kiegészíteni, hogy ezeket az elfajuló helyzeteket kiszűrjük. A program mindenesetre „rájön” (l. a levezetés 36. sorát), hogy az $ABC\triangle$ esetében kell arra vigyázni, hogy C ne essen az AB egyenesre, vagyis az esetünkben már rögzített koordinátájú A -t és B -t tekintve a C pont x -koordinátájának nem szabad 0-nak lennie. Mindez a 39. sorban az s_{ndg} egyenletben kerül feljegyzésre.

A fenti (1) egyenletünket ezek szerint az

$$(2) \quad s_1 \cdot f_1 + s_2 \cdot f_2 + s_3 \cdot f_3 + s_4 \cdot f_4 + s_5 \cdot f_5 + s_6 \cdot f_6 + s_7 \cdot f_7 + s_{ndg} \cdot f_8$$

formára egészítjük ki, és várva-várjuk, hogy a számítógép találjon nekünk olyan $f_1, f_2, \dots, f_7, f_8$ polinomokat, melyre a (2) kifejezés 1-et ad. Valóban van ilyen konfiguráció, egy ilyen lehetőség az $f_1 = -v_{13}v_{14}v_9^2$, $f_2 = v_{10}v_{13}v_{14}v_9 - v_{13}v_{14}v_9$, $f_3 = -v_{13}v_{14}^2v_5 - v_{10}v_{13}v_{14}$, $f_4 = -v_{13}v_{14}v_9$, $f_5 = -v_{13}v_9$, $f_6 = -v_{10}v_{13} + v_{13}$, $f_7 = -1$, $f_8 = -v_{10}v_{13}v_{14}v_5 + v_{13}v_{14}v_6v_9 - v_{13}v_9^2 - v_{10}^2v_{13} + v_{10}v_{13}$, ki is próbálható, bár a kézi ellenőrzés néhány percig biztosan eltart. Később részletezzük majd, hogyan találja meg a számítógép ezt a polinomegyüttest, de most megelégszünk annyival, hogy a kapott eredmény csakugyan a Thalész-tétel megfordításának egy elfogadható bizonyítása.

Azon természetesen lehet vitatkozni, hogy ez az eredmény mennyire elegáns, gyors, továbbá mennyire nem mutat semmit a tétel tartalmának geometriai hátteréről. A módszernek nem ezek az erősségei, hanem az, hogy *általánosságban*

alkalmazható³, algoritmizálható, hatékonyan számítható. Számos elemi geometriai tételre elfogadható sebességgel működik, melyek között a középiskolában (akár matematika tagozaton is) tárgyalt szinte összes geometriai állítás szerepel. Az 1. táblázat bemutat néhány prominens példát, azt is feltüntetve, hogy az $f_1, f_2, \dots$ polinomok közül a legnagyobb fokszámúnak mi a fokszáma. (A Thalész-tétel megfordításának esetében itt ez 4, de ez egyébként attól is függ, hogyan is van pontosan megfogalmazva az állítás. Egészen különböző fokszámok jöhetnek ki attól függően, hogy milyen módon alkottuk meg a szerkesztést és pontosan hogyan is szól a végkövetkeztető állítás.)

1. táblázat. Nehézségi fok és számítási igény (másodpercben) néhány középiskolában tárgyalt geometriai tétel esetében

	Tétel neve	N. fok	Sz. igény
	Thalész-tétel	2	0.019
	Thalész-tétel megfordítása (variáns)	2	0.059
	A $\triangle$ középvonala párhuzamos egy oldallal	2	0.064
	(Pitagorasz tétele)	2	0.069
	A $\triangle$ oldalfelezőinek mp.-ja a körülírható $\circ$ középpontja	2	0.073
	Feuerbach-tétel, 1. rész	3	0.079
	A $\triangle$ súlyvonalai egy pontban metszik egymást	5	0.075
	Feuerbach-tétel, 2. rész	5	0.080
	Körhöz húzott szelő- és érintőszakaszok tétele	6	0.090
	Kerületi szögek tétele	7	0.286
	Párhuzamos szelők tétele	7	0.087
	Euler-egyenes	7	0.088
	Ptolemaiosz tétele	7	0.088
	Szögfelezőtétel	7	0.110
	Menelaosz tétele	8	0.141
	Ceva tétele	8	0.814
	Wallace–Simson egyenes	16	9.144
	Feuerbach-tétel, 3. rész	18	1.352

Az 1. táblázatba felvettük a Thalész-tétel megfordításának egy olyan megfogalmazását is, ahol az AD szakaszt adjuk meg először, majd az arra merőleges (D -re illeszkedő) c egyenest, és azon jelölünk ki egy B pontot. Ilyen módon nincs szükség az elfajult eset vizsgálatára, és a nehézségi fok is csak 2 lesz. A Pitagorasz-tétel bizonyítása is csak 2-es nehézségű, de itt meg kell jegyezni, hogy a koordinátageometriai átíráskor a program impliciten alkalmazza a Pitagorasz-tételt is, tehát ez a

³Ez annyit jelent, hogy *minden* geometriai állítás, ami átfogalmazható egy racionális együtthatós polinomiális egyenletrendszerre, véges sok lépésben olyan formára hozható, ami alapján biztonsággal eldönthető, hogy az állítás *mindig igaz*, bizonyos feltételek mellett de *általánosságban igaz*, vagy *általánosságban nem igaz*.

fajta bizonyítás valójában inkorrekt, *circulus vitiosus*, ördögi kör. (A táblázatban ezért az ehhez kapcsolódó bejegyzést zárójelbe tettük.) A Feuerbach-tétel (a „9 pontos kör” tétele) bizonyításánál pedig többféle pontról is be kell lássuk, hogy azon a bizonyos körön fekszenek, és ettől függően az egyes részállítások különböző nehézségűek lehetnek.

A gyakorlatban az 5–7. fokú bizonyítások, ha kézzel nem is, de géppel viszonylag gyorsan ellenőrizhetők. Ez azt jelenti, hogy a kapott (1) vagy (2) típusú előállítást vagy a GeoGebrán belül, vagy abból kimásolva egy másik komputeralgebrai szoftverben „kiszorozzuk” (vagyis elvégezzük a polinomok szorzását: „minden tagot minden taggal”, majd egyszerűsítünk). Nagyobb nehézségi fokú állításoknál viszont rendszerint ügyeskednünk kell, mert a GeoGebrában technikai akadályokba ütközünk. Ilyenkor a helyes módszer az, hogy a CAS - komputeralgebra ablak tartalmát kiexportáljuk egy szövegfájlba, amit ezután egy „erősebb” komputeralgebrai szoftverbe (pl. Mathematicába vagy Maple-be) áttöltünk. Ez az összes említett tételre működik, habár az ellenőrzés futási ideje akár több perc is lehet.

1.3. Elimináció, Gröbner-bázis

Tartozunk még azzal, hogy eláruljuk, hogyan lehet kizárni az elfajuló eseteket, amikor a vizsgált állítás nem teljesül, valamint azzal is, hogyan talál a számítógép megfelelő $f_1, f_2, \dots$ polinomokat, melyek az (1) vagy (2) forma szerint ellentmondásra vezetnek.

A Thalész-tétel általunk vizsgált megfordításának esetében az történik, hogy azokat a $V = \{(v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11}, v_{12}, v_{13})\} \subseteq \mathbb{R}^9$ pontokat tekintjük a 9 dimenziós térben, melyekre $s_1, s_2, s_3, s_4, s_5, s_6$ és s_7 mindegyike teljesül, a kapott halmazt pedig levetítjük a (v_5, v_6) -síkra. Így olyan összefüggést (vagy összefüggéseket) kapunk v_5 és v_6 között, ami a V halmaz minden egyes pontjára érvényes.

Az olvasó lehet, hogy 9 dimenzióban kevéssé gyakorlott, ezért íme egy egyszerű példa csak 3 dimenzióban, 2 dimenzióra levetítéssel. Tekintsük a $t_1 : x \cdot y = 0$ és $t_2 : (x + y) \cdot z = 0$ egyenleteket, amelyek egyszerre teljesülnek pontosan akkor, ha $x = 0$ (ekkor y tetszőleges t_1 alapján, egyúttal $y = 0$ – és z tetszőleges – vagy $z = 0$ t_2 alapján), továbbá ha $y = 0$ (ekkor x tetszőleges t_1 alapján, egyúttal $x = 0$ – és z tetszőleges – vagy $z = 0$ t_2 alapján). A keresett halmaz tehát $(0, y, 0) \cup (0, 0, z) \cup (x, 0, 0) \cup (0, 0, z) \subset \mathbb{R}^3$, ami egyszerűsítve $(x, 0, 0) \cup (0, y, 0) \cup (0, 0, z) \subset \mathbb{R}^3$, vagyis a három tengely uniója. Ha most ezt a halmazt az (x, y) -síkra levetítjük, akkor arra jutunk, hogy az $x \cdot y = 0$ összefüggés mind az $(x, 0, 0)$, mind a $(0, y, 0)$, mind a $(0, 0, z)$ típusú pontokra helytálló lesz, ennél (t.i. az x -és y -tengely uniójánál) bővebb halmaz viszont nem adható meg, mivel egy, a tengelyekre nem eső $(a, b) \in \mathbb{R}^2$ pont $(a, b, z) \in \mathbb{R}^3$ ősképe a z -tengelyen sem lesz rajta. Vagyis a levetített halmazt az $x \cdot y = 0$ egyenlet adja meg.

A gyakorlatban az ilyen típusú kérdéseket gyakran átfogalmazzák polinom-ideálok kapcsolatos problémákká. Ebben az esetben arról van szó, hogy az

$I = \langle x \cdot y, (x + y) \cdot z \rangle$ ideált tekintjük, mely a két polinomnak bármely polinomiális $p_1 \cdot xy + p_2 \cdot (x + y)z$ kombinációját jelenti, és arra vagyunk kíváncsiak, hogy ebben a halmazban melyek azok a polinomok, amelyek nem tartalmazzák z -t. Ha rögzítjük, hogy a polinomok együtthatói milyen halmazból kerülhetnek ki (itt $\mathbb{Q}$ többnyire jó választás), akkor a feladatot úgy is megfogalmazhatjuk, hogy az $I \cap \mathbb{Q}[x, y]$ halmazt keressük, ami itt egész konkrétan $\langle xy \rangle$. Ez az eredmény amúgy jól látható I definíciójából is, mivel $(x + y) \cdot z$ bármilyen polinomiális sokszorosát véve már z is bekerül az eredménybe, tehát ezt figyelmen kívül kell hagynunk, és csakis $x \cdot y$ polinomiális sokszorosai jöhetnek szóba. Ezek pedig éppen $\langle xy \rangle$ -t alkotják.

Visszatérve a Thalész-tétel megfordítására, az $I = \langle s_1, s_2, s_3, s_4, s_5, s_6, s_7 \rangle \cap \mathbb{Q}[v_5, v_6]$ számítást végzi el helyettünk első körben a számítógép, ami a $\langle v_5, v_6 \rangle$ eredményt adja. (Az algebrai geometriában ezt a műveletet *eliminációnak* hívják.) Ez igazából *egy* elfajult esetre világít rá: ha $v_5 = 0$ és $v_6 = 0$, vagyis ha C pontosan egybeesik A -val. A programnak *elegendő* a két koordináta közül az *egyiket* (mondjuk az elsőt) 0-tól különbözőnek választania, hogy ezt az elfajult esetet elkerülje. De ez pont azt fogja jelenteni, hogy C nem esik AB -re, vagyis az $ABC\Delta$ nem elfajuló. Ezt olvassuk a levezetés 36. sorában is. Világos, hogy itt a program túl szigorú előfeltételt szabott a tétel teljesüléséhez, hiszen valójában elegendő lett volna a $C = A$ feltételt kizárni, azonban nem hibázott, mert a kimondott tétel végső soron helyes.⁴

Ezen a ponton már elméleti alapon biztosak lehetünk benne, hogy az állítás helytálló lesz, ha a megfelelő, az elfajulást megakadályozó előfeltételt (a szakzsargonban használt *non-degeneracy*-t, vagyis „ndg”-t) felvesszük a premisszák közé, és így akarunk ellentmondásra jutni. Ez úgy történik, hogy addig keresünk polinomiális összegeket valamilyen véges algoritmus segítségével, amíg az előbb-utóbb „kiköpi” azt a kombinációt, ami a konstans 1-et adja. A gyakorlatban ezt sajnos nem minden komputeralgebra rendszer tudja automatikusan⁵, mivel a legtöbbször csak az az érdekes, hogy előállítható-e az 1 vagy sem. Hogy pontosan miként, az másodlagos.

Az általában minden rendszerben rendelkezésre álló komputeralgebrai művelet az ún. *redukált minimális Gröbner-bázis* meghatározása, ami nem is mindig egyértelmű, mert különböző változósorrendek más-más eredményt adhatnak. Ha viszont a konstans 1 a fenti módon előállítható, akkor ezen komputeralgebrai művelet végeredménye minden változósorrendnél az $\{1\}$ lesz. Egy I polinomidéál esetében annak G Gröbner-bázisa – nagyon leegyszerűsítve a definíciót – I egy

⁴Az elimináció olyan eredményt is adhat, amelyben nem jelennek meg a vetítés síkjának változói. Ha a „teljes halmazt” kapjuk (vagyis, algebrai formában az $\langle 1 \rangle$ -et), akkor nincs elfajuló eset: az állítás mindig igaz. A másik lehetőség, hogy az „üres halmazt” kapjuk (ez algebrai formában $\langle 0 \rangle$, mert a 0 polinom mindig eleme a metszetnek), azt jelenti, hogy általánosságban „semmi konkrétat” nem tudunk: ilyen esetekre mondjuk, hogy az állítás általánosságban nem igaz. A köztes eset, amit mindkét példánkban látunk: az állítás általánosságban igaz – de vannak apróbb előfeltételek (ez amúgy teljesen szokványos a geometriai állításoknál).

⁵Egy működő módszer a SINGULAR [4] szoftver lift parancsának használata.

olyan részhalmazát jelenti, ami pontosan ugyanazt az ideált fedi le kombinációk sokaságával, mint I (vagyis $I = \langle G \rangle$), de a halmaz struktúrája G elemeit vizsgálva „gyorsan áttekinthető”⁶. *Redukált minimális* Gröbner-bázishoz jutunk, ha G -ből eltávolítjuk azokat az elemeket, amelyeket a többi G -beli elem polinomális kombinációjával is elő tudunk állítani. Nyilván ha a polinomideálunk redukált minimális Gröbner-bázisa $\{1\}$, akkor azonnal levonhatjuk a következtetést: az ideálnak megfelelő polinomegyenletrendszerünk bizony ellentmondásra vezet!

A redukált minimális Gröbner-bázis kiszámítására nagyon hatékony módszerek léteznek, bár hozzá kell tenni, hogy legrosszabb esetben a változók száma szerinti dupla exponenciális lépésszámmra kell felkészülni, vagyis az atomi számítások száma n változó esetén 2^{2^n} -nel arányos [5], ami minden egyes új változó esetében az előző esethez képest *négyzetes* növekedést jelent. Ez a mai számítógépek és fejlett algoritmusok használatakor még 10–20 változó esetében sem jelent problémát, de afölött már könnyen belefuthatunk olyan feladatba, ami, habár elvileg kiszámítható lenne, a gyakorlatban mégsem, mivel évekig vagy évtizedekig (vagy akár az univerzum fennállásának végén is túl) várakoznunk kellene a számítás befejeztére. Ez a gyakorlatban azt is jelenti, hogy a középiskolai matematikán messze túlmutató elemi geometriai feladatok, így például olimpiai nehézségű példák esetében nem sok esélyünk van arra, hogy a kérdéses kombinációhoz eljussunk.

Technikai szempontból megjegyezzük, hogy a kombináció pontos meghatározása lényegesen komolyabb programozási kihívás, mint a Gröbner-bázis kiszámítása, mivel a számítás lépéseinek egy komoly hányadát visszamenőleg is tárolnia kell a programnak. Ezt hatékonyan és relatíve gyorsan szükséges megtenni, különben lemondhatunk arról, hogy kivárható időn belül választ kapjunk. Másrészt az is tény, hogy egyelőre nincs olyan algoritmus, ami teljes biztonsággal ki tudná választani a legkisebb nehézségi fokú kombinációt. Jó heurisztikák vannak, de biztosíték arra, hogy a végeredmény bármilyen szempontból minimális, egyelőre nincs.

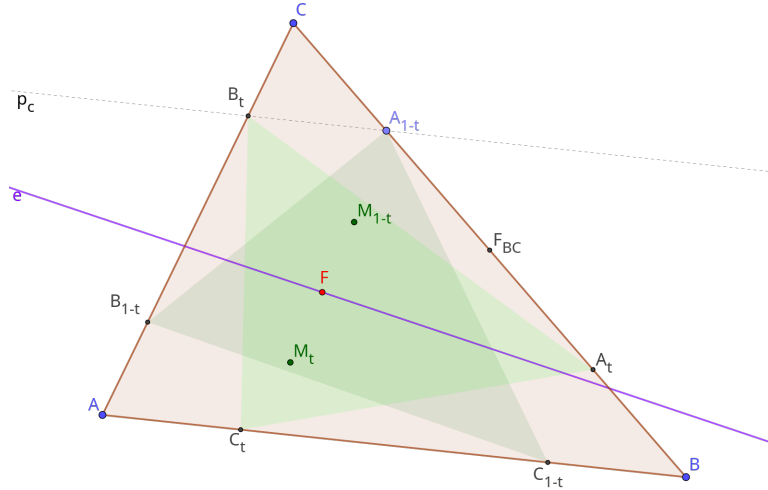
A kombináció pontos meghatározása a GeoGebra Discoveryben csak akkor történik, ha ezt külön kérjük a **BizonyításLépései** parancs hívásakor, és ez csupán 2023 óta van jelen a szoftverben. A korábban (2015 óta) rendelkezésre álló **Bizonyít** és **BizonyításRészletek** parancsok meglegészenek azzal, hogy az eliminációs lépést végrehajtsák és egy elfajulást megakadályozó előfeltételt találjanak. Ez jóval kevesebb munkával jár, ezáltal az így nyert bizonyítás sokkal gyorsabban rendelkezésre áll, mint a **BizonyításLépései** paranccsal kapott. Kevesebb munka esetén azonban nincs hitelesítés: el kell fogadnunk, hogy a gép jól számolt, vagyis nincs lehetőségünk ellenőrzésre. Ez pedig, bizonyításról lévén szó, nem biztos, hogy elfogadható kompromisszum.

A témáról további részleteket olvashatunk a [6] és [7] könyvekben, valamint a [8] és [9] cikkekben és a [10] munkában.

⁶Például azért, hogy I bármely eleme főtagjának oszthatónak kell lennie G valamely eleme főtagjával.

2. A Pete–Dötsch-tétel

A cikk hátrélvő részében felidézzük a Pete Gábor és Dötsch András által közösen talált és bizonyított tételt, melyet módszerünkkel be is bizonyítunk.



3. ábra. A Pete–Dötsch-tétel

Tekintsük a 3. ábrát! A tétel eredeti formája:

Pete–Dötsch-tétel (1998). Legyen $t \in [0, 1]$, és jelölje A_t, B_t, C_t az ABC háromszög $B, C; C, A; A, B$ csúcsai közti $1 - t, t$ súlyozással kapott pontokat.

1. Az $A_t B_t C_t$ és $A_{1-t} B_{1-t} C_{1-t}$ háromszögek magasságpontjait összekötő $M_t M_{1-t}$ szakasz felezőpontja az ABC háromszög Euler-egyenesére esik, és
2. különböző t -kre az $M_t M_{1-t}$ szakaszok egymással párhuzamosak.

Rögtön szögezzük le, hogy a tételnek csak az első részét fogjuk belátni, de az is nagy kihívás lesz. Az olvasót bátorítjuk, hogy próbálja maga igazolni a második részt.⁷

⁷Módszerünk lehetőséget ad a második rész nem hitelesített bizonyítására is, de az mintegy háromszor annyi időt igényel, mint a tétel első részének igazolása. A hitelesített bizonyításhoz a mai számítógépek szokásos erőforrásai – jelenlegi ismereteink szerint – nem elegendők.

2.1. A GeoGebra-szerkesztés

A tétel eredeti jelöléseit egy kissé kibővítjük, mivel a szerkesztés menete valamelyest más logikát fog követni, mint az eredeti bizonyításban. A háromszög csúcsait továbbra is A , B és C jelöli. Az A_{1-t} pont lesz a kiindulási pontunk, így A_t annak az F_{BC} oldalfelezőpontra tükrözött képe lesz. A B_t pontot úgy fogjuk megkapni, hogy egy p_c jelölésű egyenest húzunk az AB oldalegyenessel párhuzamosan A_{1-t} -n keresztül. Az AC oldallal való metszéspont lesz tehát B_t . Ha most az AC oldal F_{AC} felezőpontjára tükrözzük B_t -t, B_{1-t} -t kapjuk. Ha végül A_t -n át az AC oldallal húzunk párhuzamost (ezt p_b -vel jelöljük), akkor az kimetszi az AB oldalból C_{1-t} -t. Az AB oldal F_{AB} felezőpontjára tükrözve ezt C_t -t kapjuk.

Miért így közelítjük meg a feladatot, és nem a t megadásával (esetleg egy, a GeoGebra-ban jól ismert csúszkán keresztül)? A válasz rövid: csak ez a fajta szerkesztési metódus támogatott a bizonyító alrendszerben, legalábbis jelenleg.

Kijelölvén a két kis háromszög pontjait, a GeoGebra-val létrehozuk magukat a háromszögeket, melyek oldalait a $g = A_{1-t}B_{1-t}$, $h = B_{1-t}C_{1-t}$, $i = C_{1-t}A_{1-t}$, $j = A_tB_t$, $k = B_tC_t$, $l = C_tA_t$ jelölésekkel látjuk el.

Ezek után rendre definiáljuk a két kis háromszög magasságvonalait: m_h , m_i ill. m_k és m_l segítségével kapjuk meg rendre az M_{1-t} és M_t pontokat.

Hátra van még az Euler-egyenes definiálása, amelyre a következő módon kerül sor: Az $ABC\triangle$ s_a és s_b súlyvonalának metszetét S -sel, az f_a és f_b szakaszfelező merőlegeseinek metszetét O -val jelöljük. Az OS egyenes megegyezik az Euler-egyenessel (melyet itt, az eredeti publikációval összhangban e -vel jelölünk).

Végül F fogja jelölni az M_tM_{1-t} szakasz felezőpontját.

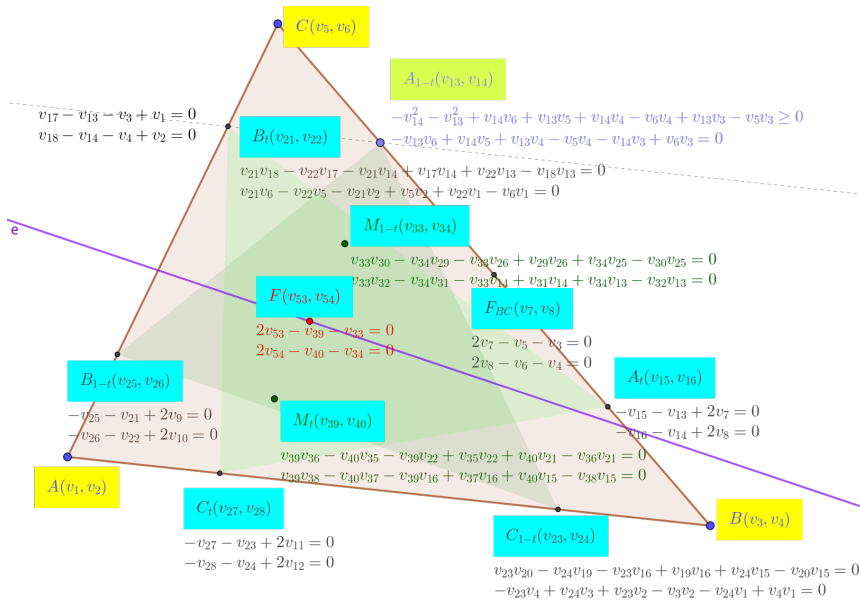
Bizonyítandó, hogy $F \in e$.

2.2. A nem hitelesített bizonyítás

Az eliminációs lépés elvégzéséig 55 változó bevezetésére van szükség, amelyet a program egy szokásos mai számítógépen mintegy 10–12 másodperc alatt végez el. A GeoGebra Discovery alapértelmezésben 5 másodperc után leállítja a számítást, tehát speciális paraméterekkel kell indítani a szoftvert, hogy eredményt érjünk el. (Erről később bővebben!)

A program sajnos arra jut, hogy csak az első pont koordinátái rögzíthetők, így csak az $A = (0, 0)$ feltételt írja fel, a másik két szabad pont (B és C) koordinátáit mindet meg kell hagyni a változók között. Ezeket v_3, v_4, v_5, v_6 jelzésekkel látja el. Az elimináció során keletkező minimális polinomhalmazból több jelölt is van, amelyeknek csupán az egyikét kiiktatva az elfajuló esetek megszüntethetők. Egy ilyen a $-v_5v_4 + v_6v_3$, melynek a jelentése, hogy az $ABC\triangle$ elfajuló. Ezt fogja kiiktatni majd a hitelesített bizonyítás; itt azonban ennyivel megelégszünk, és elfogadjuk, hogy a tétel igaz az elfajuló esetet nem számítva.

Illusztrációként a 4. ábrán figyelhetünk meg néhány részletet abból, milyen



4. ábra. A Pete–Dötsch-tételben szereplő néhány jellemző objektum algebrai alakba öltve

algebrai egyenleteket kapcsol a GeoGebra Discovery a tételben szóba jövő objektumokhoz. Nem minden egyenlet szerepel az ábrán, csak a legfontosabbak.

2.3. A hitelesített bizonyítás

Itt arra a 871 oldalas dokumentumra hivatkozunk, amelyet egy külön fájlban adunk meg: https://www.math.u-szeged.hu/polygonlap/upload/p443_1.pdf

Ennek az 1. oldala, azon belül is az 1–37. pont adja meg a szerkesztési lépéseket. Ezek lényegileg azonosak a 2.1. szakaszban említettekkel. A 38–39. sorok mondják ki, hogy az állítás igaz, ha a szerkesztés nem elfajuló, s hogy a bizonyítás indirekt módon fog történni.

A 2–4. oldalon kerül sor a szerkesztés matematikai igényű ismertetésére. A három szabad pont (A , B és C) definíciója után az F_{BC} , F_{AC} és F_{AB} oldalfelezőpontok, majd az A_{1-t} pont definíciója következik. Ez utóbbi hasonlóan van megadva, ahogyan korábban a kollinearitást kezeltük: A $BCA_{1-t}\Delta$ területére írjuk elő, hogy 0 legyen. A program jelzi, hogy az a tény, hogy egy pontot veszünk fel valamilyen szakaszon, kizárja azt a lehetőséget, hogy több szabad pont koordi-

nátáját is fixáljuk. Ennek az oka az automatikus bizonyítás óvatosságában rejlik: amennyiben egy pontot valamilyen görbén rögzítünk (itt egy egyenes szakaszon, pontosabban egy egyenesen, l. később), ahhoz egy implicit egyenletet kell a gépnek megadnia, és ebben egyik koordinátát sem tüntetheti ki előre (hogy explicit formára egyszerűsödjön az egyenlet), mivel nem tudja megjósolni, milyen forgási pozícióban lesz végül az a görbe. Éppen ezért még az AB szakasz irányát sem fixálhatja, csupán az A pont pozícióját.

A program itt egy egyenlőtlenséget is felír (57. sor), mégpedig arra vonatkozóan, hogy a G pont konkrétan a BC' szakaszon (és nem csupán az oldalegyenesen) fekszik, de ezt a későbbi számítások során nem fogja tudni kihasználni, mert a bizonyítás kizárólag polinomegyenleteken fog alapulni. (A GeoGebra Discovery bizonyos egyszerűbb, egyenlőtlenségeket tartalmazó állításokat is képes bebizonyítani, de egy másik, lényegesen lassabb módszer segítségével [11].)

Az A_{1-t} pont F_{BC} -re tükrözését a felezőpont egyenleteihez hasonlóan oldja meg a program.

A következő lépés az AB oldalegyenessel párhuzamos, az A_{1-t} pontra illeszkedő p_c egyenes megadása, amelyet egy névvel meg nem jelölt (v_{17}, v_{18}) koordinátájú segédponttal hoz létre a program. Hasonló lépésre van szükség a p_b egyenes megadásánál is.

Egészen az S pont létrehozásáig csupán a már korábban megismert egyenlet-típusokra van szükség, ami a levezetés 121. soráig tart, az e_{35} egyenletig. Ami újdonság: az ezt követő definíciók a 122–130. ill. 131–139. sorokban – ezek definiálnak 2–2 olyan pontot, amelyekre az f_a és f_b felezőmerőlegesek illeszkednek. Az egyik pont mindkét esetben a szokásos oldalfelező pont, a másik pontot pedig úgy nyerjük, hogy a kérdéses oldal valamelyik végpontját a felezőpont körül 90 fokkal valamelyik irányba elforgatjuk. Az ilyen típusú egyenletek mindig lineárisak, így komputeralgebrai szempontból nagyon stabilan kezelhetőek.

Maga a bizonyítandó állítás a 148–149. sorban tűnik fel, az $F \in e$ állítás újra egy területszámítási kérdésként jelenik meg: azt kell belátnunk, hogy az $SOF\Delta$ területe 0. Az állítás tagadó formában a 152–153. sorokban látható.

Ezután az A pontot fixáljuk (az origóba kerül).

A 2.2. szakaszban láttuk, hogy az elimináció az $ABC\Delta$ elfajulásának elkerülését javasolja, és ezt a 156–160. sorokban rögzíti is a program. Végül a 162–209. sorokban megadja, hogy a korábban összeírt 48 db egyenlet rögzített A pont mellett hogyan egyszerűsödik. A 49. egyenletet, mely az elfajulást megakadályozza, a 160. sorban már megkaptuk.

Most jön a tényleges munka: olyan polinomokat találni (49 darabot keresünk), amellyel a 49 db egyenletünket rendre megszorozva az $1 = 0$ azonosságot kapjuk, vagyis ellentmondást érünk el. A számítást a program 8–10 másodperc alatt valóban végigviszi, és a mellékelt dokumentumban is látható, hogy a 6. oldal közepétől egészen a 871. oldal közepéig, vagyis 865 oldalon át tart a kombináció ismertetése. A kombinációban megjelenő 49 db polinom maximális fokszáma 14, így az állítás

nehézségét is 14-esnek azonosítja a program.

2.4. Ellenőrzés

Van lehetőségünk arra, hogy a kapott levezetést ellenőrizzük. Ezt háromféle komputeralgebrai rendszerben is megtehetjük, ha a CAS ablakot a GeoGebra Discovery *Fájl* → *Export* menüjén keresztül kiexportáljuk. Mi itt most a Wolfram-féle `wolframscript` parancssori programot hívjuk segítségül: ez ingyenesen letölthető (regisztráció után) a <https://www.wolfram.com/wolframscript/> és <https://www.wolfram.com/developer/> címekről.

Az exportált szövegfájl végébe „kézzel” bele fogunk nyúlni: egyetlen sort fogunk hozzáírni, hogy a végső számításra is sor kerüljön. A GeoGebra Discoveryből lementett Mathematica export tartalma a következőképpen kezdődik:

```
(* Legyenek A, B, C tetszőleges pontok. *)
(* Legyen t1 az A, B, C sokszög. *)
(* Legyen c az [A, B] szakasz. *)
```

Vagyis a fájl egy része kommentekből áll, ami a szöveges feladatkiírást és a későbbi egyenletekhez fűzött magyarázatokat jelenti. Az első matematikai tartalom a 44. sorban kezdődik:

```
(* Jelöljük az F_{BC} függő pontot (v7,v8)-cal. *)
e1:=(2 * v7) - v5 - v3 == 0;
e2:=(2 * v8) - v6 - v4 == 0;
```

Egészen hasonló módon folytatódik a fájl, összesen 214 sorban. A legvégén a következőket találjuk:

```
s48:=-1 + ((v55 * v53) * v52) - ((v55 * v54) * v51) - ((v55 * v53) *
→ v42) + ((v55 * v51) * v42) + ((v55 * v54) * v41) - ((v55 * v52)
→ * v41) == 0;
(* Most tekintsük a következő egyenletet: *)
s1*(-3/4*v55^2*v56^2*v34^2*v42*v51*v22*v13*v5-3/2*v55^2*v56^2*v34*
→ v40*v42*v51*v22*v13*v5-3/4*v55^2*v56^2*v40^2*v42*v51*v22*v13*
→ v5+3/4*v55^2*v56^2*v33*v34*v42*v52*v22*v13*v5+...
(* A GeoGebra nem tudja ellenőrizni, hogy ez az 1=0-val ekvivalens,
→ de más komputeralgebra szoftver képes rá. *)
(* Ellentmondás! Ezáltal az eredeti állítást igazoltuk. *)
(* Az állítás nehézségi foka: 14. *)
```

Az utolsó egyenletnek csak az elejét vettük bele ebbe a dokumentumba, a nagy részét nem, hiszen akkor itt is 900 oldal körül járnánk ebben a pillanatban: ez a sor 5 millió karakternél is többet tartalmaz.

Most egy alkalmas szövegszerkesztővel fűzzük hozzá ehhez a szövegfájlhoz az alábbi Mathematica utasítást:

```
Timing[Print[Expand[Replace[%%%,{x_==0 :> x},2]]]]
```

Ez annyit jelent, hogy hátulról a 4. sort kezeljük úgy, mint egy szokásos polinomiális kombinációt, és értékeljük ki a zárójelek felbontásával, továbbá mérjük meg, mennyi időt vett ez igénybe.

Ha ezzel megvagyunk, már csak be kell töltsük a fájlt a `wolframscript`-be. Ha az input fájlunk neve mondjuk `melléklet.txt`, akkor – parancssorban beírva – a `wolframscript < melléklet.txt` a következőhöz hasonló kimenetet fog adni:

```
Wolfram Language 14.0.0 Engine for Linux x86 (64-bit)
Copyright 1988-2023 Wolfram Research, Inc.
```

```
In[1]:=
In[2]:=
In[3]:=
...
In[210]:=
In[211]:=
In[212]:=
In[213]:=
In[214]:=
In[215]:= 1
```

```
Out[215]= {32.9039, Null}
```

```
In[216]:=
```

A 211. sor tájékán a feldolgozás néhány másodpercre belassul: itt dolgozza fel a szoftver a hosszú inputsort, és ha végez vele, akkor utána már viszonylag gyors a hátralévő pár sornyi input végrehajtása. Az `In[215]` sor végén álló 1-es jelzi, hogy valóban 1 jön ki a polinomiális kombináció eredményeként, és pont erre van szükségünk, hogy az ellentmondást, egyúttal az indirekt bizonyítás helyességét belássuk. (A 32.9039 szám a futási időt mutatja másodpercben.)

2.5. Tippek és trükkök

E szakasz zárásaként íme néhány ötlet, ha magunk is kedvet kapnánk a kísérletezéshez ...

- A GeoGebra alapú rendszerek sokféle lehetőséget kínálnak a geometriai szerkesztés bevitelére. Ezek közül bizonyosak lehetnek teljesen támogatottak a GeoGebra Discovery bizonyító alrendszerében, máskor egyes lépések nincsenek megvalósítva – ennek lehetnek elvi és gyakorlati okai is. Például exponenciális függvényeket használó lépéseket ne adjunk meg a szerkesztésbe, hiszen

azok elvileg sem képezhetők le polinomiális egyenletekké. Ha a szokásos euklideszi körzős-vonalzós szerkesztésekben gondolkodunk, jó úton járunk, mert ezek többnyire támogatottak a bizonyító alrendszerben is.

- Ha van egy működő szerkesztésünk, de nem kapunk időben választ, próbálkozunk a szerkesztés egyszerűsítésével, más (ekvivalens) lépések használatával! A GeoGebra Discovery weboldalán a <https://github.com/kovzol/geogebra/tree/master/test/scripts/benchmark/prover/tests> címen kiindulva több száz GeoGebra fájl található, amelyek túlnyomó része hitelesítés nélkül, de nagy részük hitelesítéssel is bizonyítható. Ezekből tippeket kaphatunk, milyen szerkesztési lépésekkel érdemes dolgozni.
- Az 5 másodperces időlimitet parancssorból a `--prover=timeout:60` opcióval lehet 60 másodpercre felvinni. Ehhez a GeoGebra Discoveryt parancssorból kell indítani egy terminál ablakban, tehát ez a lehetőség a webes verzióban nem érhető el.⁸
- A CAS ablak tartalma HTML és $\text{\LaTeX}$ formátumba is kiexportálható. A mellékletben található PDF-szövegnek – kisebb kézi módosításokkal – a HTML és $\text{\LaTeX}$ változatát is csatoljuk arra az esetre, ha valaki egy saját tételt később ilyen módon szívesen publikálna. A $\text{\LaTeX}$ -változatban a GeoGebra által lementett „visszhangzott” outputokat deaktiváltuk, így a kimenet egy picivel rövidebb lett, de lényeges információ nem került törlésre.

3. Záró gondolatok

A számítógépes tételbizonyítás ma is nagy erővel hódít, talán sokkal inkább, mint 50 évvel ezelőtt, amikor Appel és Haken megtették az első komoly lépéseket a négy-szintétel igazolásának irányába [12]. A nyolcvanas évek végére százával készültek automatikus tételbizonyítások elemi geometriai tételekre, az általunk ismertetthez igen hasonló módon, elsősorban Chou munkásságának köszönhetően [7]. Az akkori algebrai jellegű bizonyítások azonban nem számítanak hitelesítettnek.

A kilencvenes években az „olvasható” bizonyítások kerültek reflektorfénybe, amelyek „emberi fogyasztásra” is alkalmasnak találtattak. Ezek módszerei inkább a formális logikai levezetésekhez állnak közel, mintsem a koordináta geometriához. Elméletüket Chou és tanítványai, Gao és Zhang dolgozták ki [13], és hoztak egyúttal nyilvánosságra hasznos szoftvereket: így a *Geometry Expert* programot, majd később annak továbbfejlesztését, a *JGEX*-et.

A dinamikus geometria szoftverek elterjedésében a GeoGebra szoftver jelentette az igazi áttörést a 2000-es évek közepén, mivel minden felhasználó számára ingyenesen hozzáférhető volt. 2010 környékén kezdődtek az első törekvések, amelyek

⁸A webes verzió is képes egyszerűbb hitelesített bizonyítások előállítására, de a Pete–Dötsch-tételnek sajnos a nem hitelesített bizonyítása is erőforrás-problémákba ütközik.

kifejezetten ebbe a programba igyekeztek automatikus tételbizonyítási módszereket beépíteni.

A versenymatematika világának örökzöld kontinensét képezik a síkgeometriai tételek bizonyítását célzó problémák. Az euklideszi geometria valóban kifogyhatatlan kincsestára újabbnál újabb fejtörő feladatoknak. Ezek szabatos bizonyítása nemcsak emberi, hanem algoritmikus szempontból is kihívás, mind a mai napig.

A mesterséges intelligenciára alapuló módszerek előretörésével új muníciót kaptak az ez irányú kutatások. Nagyon aktuális tudományos kérdés, hogy található-e olyan algoritmus, amivel nemcsak gyorsan oldhatóak meg a friss fejtörők, de egyúttal olyan módon is, hogy a megoldások az olvasó számára kevés energiabefektetéssel gyorsan és könnyen érthetőek, világosak legyenek.

Az ebben a cikkben ismertetett módszer a nyolcvanas évek technikájának egy finomított változata, különös tekintettel a levezetés részleteire, annak kommunikációjára: a bizonyítást nemcsak magyarul, hanem angolul, németül, spanyolul, franciául, olaszul, törökül, arabul és héberül is el lehet olvasni [14]. Ha a generált bizonyítás nem is a lehetséges legrövidebb, lényegesen rövidebb, mint a korábbi algebrai alapú módszer által kínált teljes levezetés, amelynél a több ezer lépésből álló gondolatmenetek sem ritkák (így azokat többnyire mellőzik).

A Pete–Dötsch-tétel esetében, valljuk be, ez a módszer továbbra is csak *egy* módszer. Az ereje mellett (ami az automatika biztonsága és pontossága) a gyengesége is előjön: elveszünk a részletekben, nem éljük át azt a fajta katarzist, amikor látjuk, pontosan min is múlik az állításnak az igazsága. Akkor jön el az automatikus bizonyítások virágkora, amikor ezt az „aha”-élményt is elő tudja majd varázsolni a számítógép. Erre a csodára azonban még várnunk kell, vagy sokkal inkább: tovább dolgoznunk érte.

Köszönetnyilvánítás

A GeoGebra Discovery bizonyító alrendszere több ember hosszú éveken keresztül intenzív munkájának az eredménye. Anélkül, hogy a fejlesztés minden fontos mérföldkövét ismertetnénk, meg kell említsük az ötletgazdát, Tomás Recio algebrai geometert, aki nyugállományba vonulása után továbbra is a projekt egyik élharcosa. A polinomiális kombináció kiszámításának hatékony C++ nyelvű implementációja a GeoGebra Discoveryben, csakúgy mint az eliminációs számoló algoritmus [15] (ez a szokásos GeoGebra-verzióban is elérhető), Bernard Parisse érdeme. Az alrendszer Java nyelvű megvalósítása, egyúttal a kimenet magyar verziójának illesztése a szerző munkája.

Irodalomjegyzék

- [1] Dötsch A., Pete G.: Egy Euler-egyeneshez kapcsolódó tételről, Polygon (Szeged) VIII:1 (1998), 73-80.

- [2] Tarcsay T.: Születnek-e napjainkban is geometriai tételek? Sulinet Hírmagazin, 2002/12/09. <https://hirmagazin.sulinet.hu/hu/pedagogia/szuletn-ek-e-napjainkban-is-geometriai-tetelek>
- [3] Rabinowitsch, J.L.: Zum Hilbertschen nullstellensatz, Math. Ann., 102(1) (1929), 520.
- [4] Decker, W., Greuel, G.-M., Pfister, G., Schönemann, H.: SINGULAR 4-4-0 — A computer algebra system for polynomial computations. <https://www.singular.uni-kl.de> (2024).
- [5] Mayr, E., Meyer, A.: The complexity of the word problem for commutative semigroups and polynomial ideals, Advances in Mathematics 46 (1982), 305-329.
- [6] Cox, D., Little, J., O’Shea, D.: Ideals, varieties, and algorithms, Springer New York (2007)
- [7] Chou, S.C.: Mechanical geometry theorem proving, Springer Science + Business Media (1987)
- [8] Kapur, D.: Using Gröbner bases to reason about geometry problems, Journal of Symbolic Computation 2 (1986), 399-408.
- [9] Kovács, Z., Recio, T., Vélez, M.P.: Showing proofs, assessing difficulty with GeoGebra discovery, In: Quaresma, P., Kovács, Z. (eds): Proceedings 14th International Conference on Automated Deduction in Geometry, Belgrade, Serbia, 20-22th September 2023, Electronic Proceedings in Theoretical Computer Science 398 (2024), 43-52.
- [10] Kovács, Z.: Computer Based conjectures and proofs in teaching Euclidean geometry, PhD thesis, Johannes Kepler University, Linz (2015)
- [11] Brown, C.W., Kovács, Z., Luksch, S., Recio, T., Vajda, R., Vélez, M.P.: Towards detection of partial truth via real geometry, In: Ábrahám, E., Sturm, T. (eds): Proceedings of the 8th International Workshop on Satisfiability Checking and Symbolic Computation, July 28, 2023, Tromsø, Norway, Collocated with ISSAC 2023 (Tromsø, Norway). CEUR Workshop Proceedings 3455 (2023). <https://ceur-ws.org/Vol-3455/short2.pdf>
- [12] Appel, K., Haken, W.: The four-color problem, In: Steen, L.A. (ed): Mathematics today twelve informal essays (1978), 153-180.
- [13] Chou, S.C., Gao, X.S., Zhang, J.Z.: Automated generation of readable proofs with geometric invariants, Journal of Automated Reasoning 17 (1996), 349-370.

- [14] Kovács, Z., Parisse, B., Recio, T., Vélez, M.P., Yu, J.H.: The `ShowProof` command in *GeoGebra Discovery*: Towards the automated ranking of elementary geometry theorems, *ACM Communications in Computer Algebra* 58(2) (2024), 27-30.
- [15] Kovács, Z., Parisse, B.: Giac and GeoGebra – Improved Gröbner Basis Computations, In: Gutierrez, J., Schicho, J., Weimann, M. (eds): *Computer Algebra and Polynomials* (2015), 126-138.

Kovács Zoltán, PHDL, Linz, Ausztria