



# Bizonyítás az emberin túl

Bizonyítás az emberin túl, a mesterséges intelligencián innen

---

Molnár Zoltán Gábor

2024. október 18.

Matematika Intézet  
BME TTK

# Miről lesz szó?

1. Következtetések természetes nyelven
2. Ezek számítógépes szabályai
3. A példák ellenőrzése a Coq (Thierry Coquand, 1989, INRIA)
4. Példa az alkalmazására
5. Érti-e a matekot a bizonyításellenőrző?
6. Érti-e a matekot az ember?

Árverésre bocsátom az alábbi *következtetéseket*.

(Az "érvényes" csapat az érvényesekre hajtson, az "érvénytelen" csapat az érvénytelenekre.)

**1.**

Ha Anna könyvtáros, akkor Anna csendes.

—

**Tehát:**

Anna nem könyvtáros, vagy Anna csendes.

**2.**

Ha Anna könyvtáros, akkor Anna csendes.

Ha Anna csendes, akkor Anna okos.

—

**Tehát:**

Ha Anna könyvtáros, akkor Anna okos.

**3.**

Peti tanár, feltéve, hogy ha Peti hangos,  
akkor Peti tanár.

—

**Tehát:**

Peti tanár.

**4.**

Ha Peti tanár, akkor Peti hangos.  
Peti hangos.

—

**Tehát:**

Peti tanár.

## Bevezetési szabály

Legyen az  $ABC\triangle$ -nek a  $C$ -nél lévő szöge  $90^\circ$ , ekkor ..., így  $a^2 + b^2 = c^2$ .

Tehát, ha az  $ABC\triangle$ -nek a  $C$ -nél lévő szöge  $90^\circ$ , akkor  $a^2 + b^2 = c^2$ .

$$\frac{\begin{array}{c} [x : A] \\ \vdots \\ t : B \end{array}}{\lambda x. t : A \rightarrow B} \text{ intro}$$

## Kiküszöbölési szabály

Ha  $A$ , akkor  $B$ . De  $A$ . Tehát  $B$ .

$$\frac{f : A \rightarrow B \quad a : A}{f a : B} \text{ apply } f$$

## A nyelvi jelentés használatelmélete

A *használatelmélet* szerint a szavak jelentése nem az által adott, hogy milyen dolgokra utalnak, vagy milyen agyi képek kapcsolódnak hozzájuk, hanem azáltal, hogy a kommunikációban hogyan használjuk őket.

A *logikai* műveletek értelme szintén megadható így. Sőt, matematikai és elméleti számítástudományi fogalmaké is, a módszer skálázódik.

A *mondatműveletek használatelméleti jelentését* az adja, hogy

1. **verifikációs jelentésrész:** milyen **feltételek** teszik igazzá a mondatot, amiben szerepelnek
2. **pragmatikus jelentésrész:** mely állítások a **következményeik**, mire lehet következtetni belőlük.

## Bevezetési szabály

$$\frac{p : A \quad q : B}{\text{conj } p \ q : A \wedge B} \text{ split}$$

## Kiküszöbölési szabály (destruktor)

$$\frac{p : A_1 \wedge A_2}{p_i : A_i} \quad (i : 1; 2) \text{ destruct } p \text{ as } [p_1 \ p_2]$$

## Bevezetési szabály

$$\frac{p : A_i}{\text{incl}_i A_1 \vee A_2} \quad (i : 1; 2) \text{ left/right}$$

## Kiküszöbölési szabály (esetszétválasztás, switch)

$$[x : A] \quad [y : B]$$

$$\frac{p : A \vee B \quad q : C \quad r : C}{\text{orelim}(p, x.q, y.r) C} \quad \text{destruct } p \text{ as } [x|y]$$

# Következtetési szabályok, összefoglalás

bev. szabály	Coq taktika	kik. szabály	Coq taktika
$\frac{[A] \quad \vdots \quad B}{A \rightarrow B}$	intro	$\frac{A \rightarrow B \quad A}{B}$	apply
$\frac{A \quad B}{A \wedge B}$	split	$\frac{A_1 \wedge A_2}{A_i}$	destruct
$\frac{A_i}{A_1 \vee A_2}$	left/right	$\frac{A \vee B \quad [A] \quad [B] \quad \vdots \quad \vdots \quad C \quad C}{C}$	destruct
$\sim A = A \rightarrow \text{False}$			



## Bizonyítás asszisztensek 1.

- Coq (Thierry Coquand, 1989) <https://coq.inria.fr/> (Coq-ban)
- Lean (Leonardo de Moura, 2013) <https://lean-lang.org/> (C++-ban)
- Agda (Ulf Norell; Catarina Coquand, 1999) <https://github.com/agda/agda> (Haskell-ben)
- Idris (Edwin Brady, 2007) <http://docs.idris-lang.org/en/latest/> (Haskell-ben)

## Bizonyítás asszisztensek 2.

### Amit **tudnak**:

- az ember és a gép együttműködésére építenek
- interaktív felületen az ember irányítja a bizonyítás keresését
- ellenőrzi a betáplált formális bizonyításokat
- gép tárolja a részleteket és javasol bizonyításkeresési utakat
- új fejlesztési irány az eszközök AI kiegészítése

### Amit **nem tudnak**:

- nincs *általános* eljárás, amely bármely formalizált matematikai állításról eldönti, hogy igaz vagy hamis (Church-tétel)

### Példák (github)

- az elárvezett mondatok Coq-megoldása (.v)
- halmazeméleti feladat Coq-megoldása (.v)
- aritmetikai feladat Coq-megoldása (.v)
- számelméleti feladat Lean megoldása (.lean)

**John Searle** (1932-) gondolkísérlete, a *kínai szoba*.

Egy ember, aki nem tud kínaiul, képes lehet helyes válaszokat adni kínai kérdésekre pusztán a rendelkezésére álló nagyon részletes szabálykönyv alapján.



Szimbólummanipulációt végez a számítógép is. Ha az ember nem érti miről szól a beszélgetés, vajon a gép hogy értené?

A számítógép nem érti annyira a nyelvet mint mi (de legalább is csak legfeljebb annyira értheti.)

**Michael Dummett** (1925-2011) szerint ha lennének nem kommunikálható matematikai tartalmak, akkor azok olyan matematikai tételek lennének, amik nem tehetők semmilyen módon közkinccsé. "Csodálatos bizonyítást találtam, de kevés a margó, semhogy befogadná." (Fermat)



A proof assistant-ekbe pontosan azokat a szabályokat programozzuk be, amik a bizonyításokat igazzá teszik (logikai szabályok, matematikai axiómák). Ha lenne olyan szabály, amit nem tudnánk leprogramozni, akkor arról egymást se tudnánk meggyőzni. Ezért az ember valójában nem értheti jobban a matekot, mint a számítógép.

```
Fixpoint összeg (n:nat) :=
match n with
| 0 => 0
| S n => (összeg n) + S n
end.

Theorem első_n_szám_összege : forall n, 2*(összeg n) = n*(n+1).
Proof.
intros.
induction n.
simpl.
reflexivity.
simpl.
simpl in IHn.
lia.
Show Proof.
Qed.
```

Tehát vagy az van, hogy a **matekot** se az ember, se a számítógép nem érti, vagy az van, hogy mindkettő érti.