A Kepler sejtés bizonyításának ellenőrzése

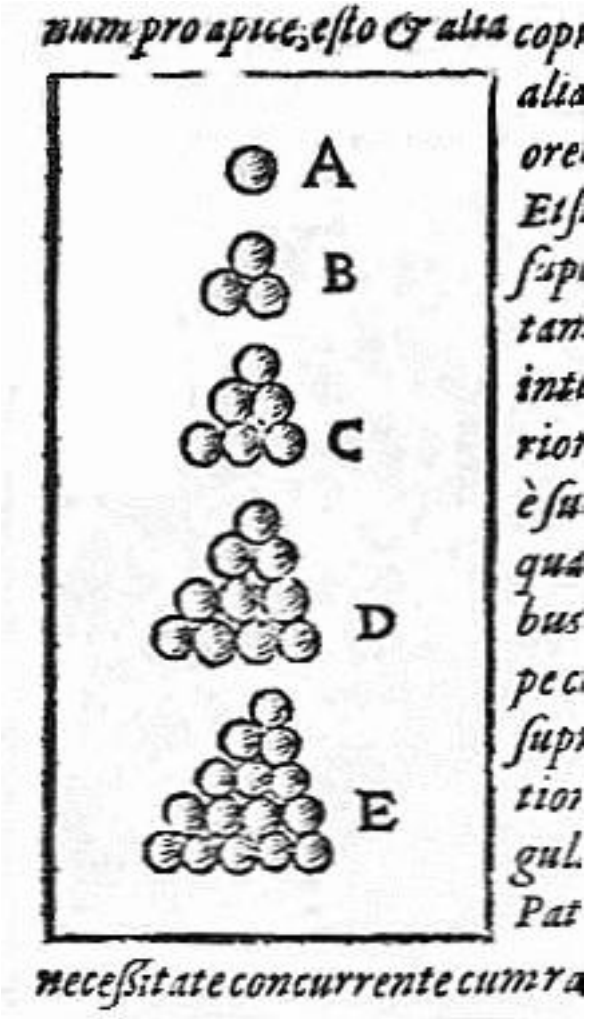
HOL Light és Isabelle segítségével

(Flyspeck projekt, 2014)

Kovács István, doktorandusz, IIT

Kepler sejtés

- Sejtés: Gömbökkel a legjobb térkitöltés a hexagonális és a lapközepes kockaráccsal érhető el. (Johannes Kepler, 17. század)
- Teljes bizonyítás: Thomas Hales, Annals of Mathematics, 2005
- A bizonyítás nagy része számítógép segítségével készült (többváltozós magasabbrendű egyenlőtlenség-rendszerek megoldása)
- Az alapötlet (bizonyítási stratégia) Fejes Tóth Lászlótól származik



A bizonyítás

- A teljes bizonyítás több mint 300 oldal
- A referee csoport csak 99%-ig volt biztos a helyességében
- Flyspeck projekt: 2014-re a teljes bizonyítást formalizálták és tételellenőrző szoftverekkel ellenőrizték
- Bizonyítási stratégia:
  - Bontsuk fel a teret korlátos (kicsi) cellákra (Marchal cellákra)
  - Indirekt: tegyük fel, hogy van egy ellenpélda, ekkor erre nagyon sok feltételnek teljesülnie kell
  - Egy cellán belül nem túl sok gömb lehet, esetszétválasztás aszerint, hogy a középsőt hány érinti
  - Az egyes ellenpéldákra teljesülő nemlineáris egyenlőtlenségeket lineáris programmá alakítjuk, amikről bizonyítható az ellentmondásosság
  - Végül: az összes lehetséges ellenpélda ellentmondásra vezet

Idézetek

"The news from the referees is bad, from my perspective. They have not been able to certify the correctness of the proof, and will not be able to certify it in the future, because they have run out of energy to devote to the problem. This is not what I had hoped for."

(Robert MacPherson)

"The referees put a level of energy into this that is, in my experience, unprecedented."

(Thomas Hales)

HOL Light

- HOL (Higher Order Logic) tételbizonyító család tagja
- Típuselméleten alapul
- Interaktív tételbizonyítás
  - Nem kell beírni a teljes formális bizonyítást, sok következtetési szabályt automatikusan használ
  - A tételek bizonyítása interaktív „játék”
- OCaml alapú
  - Funkcionális programnyelv
  - Sok hasznos segédeszközt tartalmaz (pl lexer)
- A tételek olyan típusok, amik csak az axiómákból jöhetnek ki következtetési szabályok segítségével
- Maga a kernel csak pár száz soros kód
- Linux, OS X, Windows támogatott

HOL Light, formalizált bizonyítások

- Gödel első nemteljességi tétele
- Prímszámtétel
- Jordan görbe tétel
- Kvadratikus reciprocitási tétel
- Algebra alaptétele
- Pithagorasz tétel
- ... még kb. 100 egyéb kisebb tétel

HOL-Light, szintaxis

- ∀ univerzális kavantor: `!`
- ∃ egzisztenciális kvantor: `?`
- Logikai szimbólumok: és `/\`, vagy `\/`, konjunkció `==>`
- Tétel: `|-`
- Valós számok: `&` (pl. `&3` az 3.0 valós számot jelöli, de a `3` a 0 harmadik rákövetkezőjét)
- Példa: `|- ! &x ( x < &0 \/ &0 <= x )`

HOL-Light eszközök

$\dfrac{}{\vdash t = t}$	REFL	reflexivity of equality
$\dfrac{\Gamma \vdash s = t \qquad \Delta \vdash t = u}{\Gamma \cup \Delta \vdash s = u}$	TRANS	transitivity of equality
$\dfrac{\Gamma \vdash f = g \qquad \Delta \vdash x = y}{\Gamma \cup \Delta \vdash f(x) = g(y)}$	MK_COMB	congruence of equality
$\dfrac{\Gamma \vdash s = t}{\Gamma \vdash (\lambda x.s) = (\lambda x.t)}$	ABS	abstraction of equality (x must not be free in Γ)
$\dfrac{}{\vdash (\lambda x.t)x = t}$	BETA	connection of abstraction and function application
$\dfrac{}{\{p\} \vdash p}$	ASSUME	assuming p, prove p
$\dfrac{\Gamma \vdash p = q \qquad \Delta \vdash p}{\Gamma \cup \Delta \vdash q}$	EQ_MP	relation of equality and deduction
$\dfrac{\Gamma \vdash p \qquad \Delta \vdash q}{(\Gamma - \{q\}) \cup (\Delta - \{p\}) \vdash p = q}$	DEDUCT_ANTISYM_RULE	deduce equality from 2-way deducibility
$\dfrac{\Gamma[x_1, \ldots, x_n] \vdash p[x_1, \ldots, x_n]}{\Gamma[t_1, \ldots, t_n] \vdash p[t_1, \ldots, t_n]}$	INST	instantiate variables in assumptions and conclusion of theorem
$\dfrac{\Gamma[\alpha_1, \ldots, \alpha_n] \vdash p[\alpha_1, \ldots, \alpha_n]}{\Gamma[\tau_1, \ldots, \tau_n] \vdash p[\tau_1, \ldots, \tau_n]}$	INST_TYPE	instantiate type variables in assumptions and conclusion of theorem

HOL-Light további tulajdonságai

- Számos matematikai fogalmat ismer: mérték, geometriai jelölések, térbeli testek tulajdonságai, affin és konvex halmazok a térben
- Számos előre bizonyított tételt tartalmaz: Brouwer fix-pont tétel, Krein-Milman tétel, Stone-Weierstrass tétel és számos ekevésbé ismert de hasznos lemma
- Használató az “az általánosság megsértése nélkül” típusú érvelés

Isabelle

- HOL (Higher Order Logic) tételbizonyító család tagja
- Zermelo–Fraenkel set theory (ZFC)
- Többféle stílusban írhatunk bizonyításokat
- Számos informatikai felhasználás
  - HP 9000 szerver előzetes tesztelése, szimulálása
  - seL4 mikokernel hibamentességének bizonyítása (a projekt során 160 hibát találtak!)
  - Lightweight Java típusbiztonsága
- Linux, OS X, Windows támogatott

```
theorem sqrt2_not_rational:
  "sqrt (real 2) ∉ ℚ"
proof
  assume "sqrt (real 2) ∈ ℚ"
  then obtain m n :: nat where
    n_nonzero: "n ≠ 0" and sqrt_rat: "¦sqrt (real
2)¦ = real m / real n"
    and lowest_terms: "gcd m n = 1" ..
  from n_nonzero and sqrt_rat have "real m = ¦sqrt
(real 2)¦ * real n" by simp
  then have "real (m²) = (sqrt (real 2))² * real
(n²)" by (auto simp add: power2_eq_square)
  also have "(sqrt (real 2))² = real 2" by simp
  also have "... * real (m²) = real (2 * n²)" by
simp
  finally have eq: "m² = 2 * n²" ..
  hence "2 dvd m²" ..
  with two_is_prime have dvd_m: "2 dvd m" by (rule
prime_dvd_power_two)
  then obtain k where "m = 2 * k" ..
  with eq have "2 * n² = 2² * k²" by (auto simp add:
power2_eq_square mult_ac)
  hence "n² = 2 * k²" by simp
  hence "2 dvd n²" ..
  with two_is_prime have "2 dvd n" by (rule
prime_dvd_power_two)
  with dvd_m have "2 dvd gcd m n" by (rule
gcd_greatest)
  with lowest_terms have "2 dvd 1" by simp
  thus False by arith
qed
```

Tételbizonyítók összehasonlítása

proof assistant	*proof style of the system*
HOL Light	procedural
Mizar	declarative
ProofPower	procedural
Isabelle	both possible
Coq	procedural

Procedural: a bizonyítás fő lépéseit, céljait adjuk meg, a maradékot automatikusan generálja a program

Declarative: a bizonyításokat egy az egyben apró lépésenként adjuk meg (pl. programkódok ellenőrzésénél használják)

Flyspeck projekt

- Thomas Hales vezette
- Célja a Kepler sejtés bizonyításának teljes formalizálása és verifikációja
- 2014-ben sikeresen lezárták
- 500,000 sornyi script
- Bárki számára hozzáférhető:
  https://code.google.com/p/flyspeck/
  http://arxiv.org/abs/1501.02155v1

A Kepler sejtés formalizálása

Legyenek a gömbök egységgömbök

```
|- packing V <=>
  (!u v. u IN V /\ v IN V /\ dist(u,v) < &2 ==> u = v)
```

Pakolás: bármely két gömb diszjunkt, azaz a középpontjaik távolsága legalább 2

```
|- the_kepler_conjecture <=>
    (!V. packing V
       ==> (?c. !r. &1 <= r
          ==> &(CARD(V INTER ball(vec 0,r))) <=
             pi * r pow 3 / sqrt(&18) + c * r pow 2))
```

Minden pakolásra létezik c, hogy bármely r sugarú gömbre, a gömbön belüli pontok száma:

$$\frac{\pi r^3}{\sqrt{18}} + c r^2$$

A Kepler sejtés bizonyítása

- Több ezer nemlineáris egyenlőtlenség: `the_nonlinear_inequalities`
- A lehetséges ellenpéldák kombinatorikus struktúrája: `import_tame_classification`
- Lineáris programok: `linear_programming_results`
- A Kepler sejtés:

```
|- the_nonlinear_inequalities /\
     import_tame_classification ==>
     the_kepler_conjecture
```

A verifikáció

- `the_nonlinear_inequalities`
  - Microsoft Azure felhőben 5000 processzor óra
  - Radboud University: 60 hyperthreading Xeon 2.3GHz CPUs: 6 nap
- Ezek ismeretében egy átlagos számítógépen kb. 4 óra a verifikáció

Hivatkozások

- *Johannes Kepler,* Six-Cornered Snowflake, 1611
- *Freek Wiedijk,* Formal Proof—Getting Started
- *Thomas Hales et al.,* The formal proof of the Kepler conjecture, 2014
- Wikipédia (HOL Light, Isabelle, Kepler conjecture)