Grumpy Old Man

John Harrison, Intel Corporation

Workshop on Interactive Theorem Proving
Cambridge

Mon 24th August 2009 (14:15 – 14:30)

Summary of main points

- Top-down development is a bad idea
- Modular theories are a waste of time
- Automation is a hindrance

Top-down development

How can we expect to get intermediate statements right?

- We need the ‘sanity check’ involved in proving them: they might be false.
- We need the ‘sanity check’ involved in using them: they might be too weak.

The former problem seems worst because we want to avoid relying on false statements.

Examples

- Every set of reals that is bounded above has a least upper bound

$$\forall S.\ \ (\exists M.\, \forall x \in S.\, x \leqslant M)$$
$$\Rightarrow \exists m.\, (\forall x \in S.\, x \leqslant m) \wedge (\forall m' < m.\, \exists x \in S.\, x > m'))$$

- The surface of a circular cone

$$\mathsf{cc}(v, w, r) = \{x \mid (x - v) \cdot w = \|x - v\|\|w\|r\}$$

has measure zero

$$\forall v, w, r.\, \mathsf{NULLSET}(\mathsf{cc}(v, w, r))$$

Examples

- Every set of reals that is bounded above has a least upper bound

$$\forall S.\ \ (\exists x.\, x \in S) \wedge (\exists M.\, \forall x \in S.\, x \leqslant M)$$
$$\Rightarrow \exists m.\, (\forall x \in S.\, x \leqslant m) \wedge (\forall m' < m.\, \exists x \in S.\, x > m'))$$

- The surface of a circular cone

$$\mathsf{cc}(v, w, r) = \{x \mid (x - v) \cdot w = \|x - v\|\|w\|r\}$$

has measure zero

$$\forall v, w, r.\, \mathsf{NULLSET}(\mathsf{cc}(v, w, r))$$

Examples

- Every set of reals that is bounded above has a least upper bound

$$\forall S.\ \ (\exists x.\, x \in S) \wedge (\exists M.\, \forall x \in S.\, x \leqslant M)$$
$$\Rightarrow \exists m.\, (\forall x \in S.\, x \leqslant m) \wedge (\forall m' < m.\, \exists x \in S.\, x > m'))$$

- The surface of a circular cone

$$\mathsf{cc}(v, w, r) = \{x \mid (x - v) \cdot w = \|x - v\|\|w\|r\}$$

has measure zero

$$\forall v, w, r.\, \neg(w = 0) \Rightarrow \mathsf{NULLSET}(\mathsf{cc}(v, w, r))$$

Modular theories

Much attention has been given to developing modular ‘little theories’ to structure proofs, proving results in their most general form and instantiating them.

Sometimes this can be very valuable, but often it just adds intellectual load.

In much of mathematics we usually want to assume some standard ‘big theory’.

Modular theories

The prelude of the HOL Light theory of analysis in Euclidean space:

```
loadt "Examples/card.ml";;
loadt "Examples/permutations.ml";;
loadt "Multivariate/misc.ml";;
loadt "Examples/products.ml";;
loadt "Examples/floor.ml";;
loadt "Examples/binomial.ml";;
loadt "Examples/iter.ml";;
```

Why are they all used?

A typical Mizar proof

```
:: Euler
theorem
  n0 is even & n0 is perfect implies
  ex p being natural number
  st 2|^p -' 1 is prime & n0 = 2|^(p -' 1)*(2|^p -' 1)
proof
  assume n0 is even; then
  consider k',n' be natural number such that
  A1: n' is odd & k' > 0 & n0 = 2|^k' * n' by Th2;
  assume n0 is perfect; then
  A2: sigma n0 = 2 * n0 by Def6;
  k'>=0+1 by A1,NAT_1:13; then
  A3: k'+1 >= 1+1 by XREAL_1:9;
  set p=k'+1;
  A4: p - 1 = p -' 1 by XREAL_0:def 2;
  take p;
  reconsider n2=n' as non zero natural number by A1;
  A5: 2|^p - 1 > p - 1 by NEWTON:105,XREAL_1:16;
  A6: 2|^p - 1 = 2|^p -' 1 by A5,XREAL_0:def 2;
  sigma(2|^(p -' 1)) = (2|^(p -' 1 + 1) - 1)/(2 - 1) by INT_2:44,Th30
```

A typical Mizar proof (continued)

```
  .= 2|^p - 1 by A4; then
  A7: (2|^p - 1)*sigma(n') = 2 * 2|^(p -' 1)*n' by A4,Th37,A1,Th3,A2
  .= 2|^p*n' by A4,NEWTON:11; then
  A8: (2|^p -' 1) divides 2|^p*n2 by A6,INT_1:def 9;
  2|^p - 1 = 2|^(p-'1+1)-2+1 by A4
  .= 2|^(p-'1)*2-2+1 by NEWTON:11
  .= 2*(2|^(p-'1)-1)+1; then
  (2|^p -' 1) divides n2 by A8,EULER_1:14,A6,Th3; then
  consider n'' be natural number such that
  A9: n' = (2|^p -' 1)*n'' by NAT_D:def 3;
  A10: n'' divides n' by A9,NAT_D:def 3;
  2|^p > 2 by NEWTON:105,A3,XXREAL_0:2; then
  2|^p -1 > 2-1 by XREAL_1:16; then
  A11: (2|^p -' 1)*n2 > 1*n2 by A6,XREAL_1:70;
  sigma(n')*(2|^p - 1) = 2|^p*n''*(2|^p - 1) by A6,A9,A7; then
  sigma(n2) = (2|^p - 1)*n'' + n'' by A5,XCMPLX_1:5
  .= n' + n'' by A5,XREAL_0:def 2,A9; then
  A12: n''=1 & n' is prime by Th33,A10,A11,A9;
  hence 2|^p -' 1 is prime by A9;
  thus n0 = 2|^(p -' 1)*(2|^p -' 1) by A9,A12,A1,A4;
end;
```

The imports etc. for this file

```
environ
 vocabularies ORDINAL2, ARYTM, FINSEQ_1, ARYTM_3, ARYTM_1, RELAT_1, FUNCT_1,
      BOOLE, QC_LANG1, CARD_3, FINSET_1, XREAL_0, GROUP_1, NAT_1, INT_1,
      FILTER_0, CARD_1, TARSKI, SQUARE_1, POWER, EULER_1, MATRIX_2, INT_5,
      ABSVALUE, GR_CY_1, COMPLEX1, MOEBIUS1, BHSP_5, NAT_5, FUNCT_2, RLVECT_1,
      SUPINF_1, FUNCOP_1, FINSEQ_2, WAYBEL29, TOPGEN_1, COHSP_1, POLYNOM1,
      UPROOTS, ALGSEQ_1, MONOID_0, RFINSEQ, PARTFUN1, RFUNCT_3, INT_2, NAT_3,
      NAT_LAT, XXREAL_2, MEMBERED, PROB_1;
 notations VALUED_1, TARSKI, XBOOLE_0, ZFMISC_1, SUBSET_1, SETFAM_1, FUNCOP_1,
      RELAT_1, FUNCT_1, RELSET_1, PARTFUN1, MCART_1, FUNCT_2, ORDINAL1,
      NUMBERS, CARD_3, CARD_1, XCMPLX_0, XREAL_0, FINSEQ_1, FINSEQ_2, FINSEQ_3,
      INT_1, INT_2, NAT_1, NAT_D, RVSUM_1, REAL_1, SQUARE_1, XXREAL_0, NEWTON,
      ABIAN, EULER_2, PEPIN, ABSVALUE, EQREL_1, INT_5, MEMBERED, FINSEQOP,
      CARD_FIN, ENUMSET1, FINSET_1, COMPLEX1, NAT_3, DOMAIN_1, POWER, MOEBIUS1,
      INT_3, BHSP_5, EULER_1, WSIERP_1, BINOP_1, PROB_3, RECDEF_1, SUPINF_1,
      CONVFUN1, POLYNOM1, UPROOTS, BINOP_2, FUNCT_3, RFINSEQ, RFUNCT_3,
      XXREAL_2, CLASSES1;
```

The imports etc. for this file (continued)

```
constructors VALUED_1, RELAT_2, PARTFUN1, MCART_1, SETFAM_1, FUNCT_2,
     WELLORD2, REAL_1, SQUARE_1, NAT_1, NAT_D, BINOP_2, INT_2, FINSOP_1,
     RVSUM_1, NEWTON, WSIERP_1, ABIAN, EULER_1, EULER_2, PEPIN, ABSVALUE,
     EQREL_1, INT_4, ZFMISC_1, RELSET_1, INT_5, RECDEF_1, MEMBERED, SEQ_4,
     MOEBIUS1, FINSEQOP, NUMBERS, MESFUNC2, CONVFUN1, CARD_FIN,
     XXREAL_0, COMPLEX1, INT_1, TARSKI, ENUMSET1, FUNCOP_1, XREAL_0, CARD_1,
     XCMPLX_0, UPROOTS, NAT_3, REALSET1, FUNCT_1, POWER, INT_3, BHSP_5,
     FINSEQ_1, FINSEQ_5, RFINSEQ, CALCUL_2, BINOP_1, PROB_3, SUPINF_1,
     FINSEQ_3, GOBRD10, POLYNOM1, FUNCT_3, RFUNCT_3, VALUED_0, XXREAL_2,
     CLASSES1, PBOOLE;
registrations XBOOLE_0, ORDINAL1, RELSET_1, FINSET_1, NUMBERS, XCMPLX_0,
     XXREAL_0, XREAL_0, NAT_1, INT_1, INT_2, MEMBERED, FINSEQ_1, FINSEQ_2,
     RVSUM_1, VALUED_0, NEWTON, ABIAN, PEPIN, ABSVALUE, EQREL_1, INT_4,
     FUNCT_1, ZFMISC_1, SUBSET_1, INT_5, MOEBIUS1, FUNCT_2, PRE_CIRC,
     CARD_FIN, DYNKIN, RELAT_1, PARTFUN1, SQUARE_1, CARD_1, PREPOWER, POWER,
     INT_3, BHSP_5, WSIERP_1, FINSEQ_5, FINSEQ_7, RFINSEQ, CALCUL_2, BINOP_1,
     SUPINF_1, CONVFUN1, POLYNOM1, NAT_3, UPROOTS, BINOP_2, FUNCT_3, XXREAL_2,
     CLASSES1;
requirements REAL, NUMERALS, SUBSET, BOOLE, ARITHM;
```

The imports etc. for this file (continued)

```
definitions TARSKI, XBOOLE_0, XXREAL_0, NAT_1, NAT_D, RELAT_1, FUNCT_1, INT_1,
     INT_2, INT_5, NEWTON, FINSEQ_1, FINSEQ_3, RVSUM_1, ORDINAL1, ABIAN,
     EULER_1, ABSVALUE, MOEBIUS1, FUNCT_2, CLASSES1;
theorems XBOOLE_0, XBOOLE_1, XXREAL_0, XREAL_1, XCMPLX_1, NAT_1, NAT_D,
     RELAT_1, FUNCT_1, INT_1, INT_2, INT_4, INT_5, NEWTON, EULER_1, EULER_2,
     FINSEQ_1, FINSEQ_2, FINSEQ_3, FINSEQ_5, RVSUM_1, WSIERP_1, ORDINAL1,
     IDEA_1, INT_6, INT_3, FIB_NUM3, POWER, PEPIN, SQUARE_1, CARD_1, CARD_2,
     ZFMISC_1, ALGSEQ_1, FINSEQ_4, JGRAPH_1, TARSKI, COMPLEX1, ABSVALUE,
     MOEBIUS1, CARD_3, MESFUNC3, FUNCT_2, CONVFUN1, FINSEQOP, FUNCOP_1,
     BAGORDER, RELSET_1, WELLORD2, PARTFUN1, BHSP_5, UPROOTS, RFINSEQ,
     BINOP_2, BINOP_1, POLYNOM1, FUNCT_3, RFUNCT_3, NAT_4, NAT_3, RADIX_1,
     XREAL_0, ABIAN, ENUMSET1, MONOID_1, CLASSES1;
schemes NAT_1, FUNCT_1, FUNCT_2, BINOP_2, FINSEQ_1;
```

Automation is a hindrance

- Encourages an abrogration of understanding, not realizing the ideal combination.
- Encourages inefficient work ‘massaging’ the problem until it is solved automatically.
- Standard decision procedures are often not what’s needed.

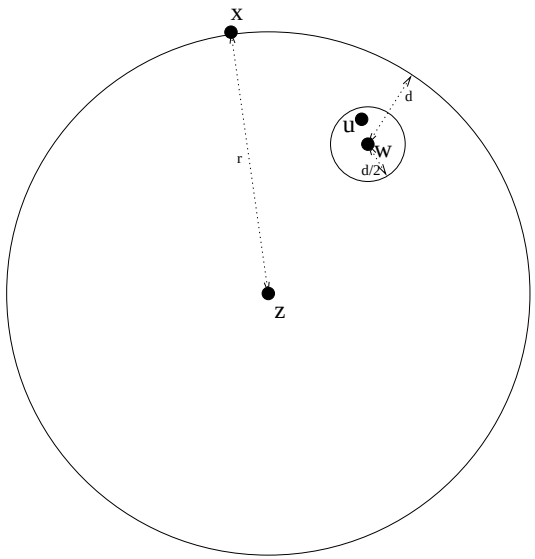
Automation gap in formalizing number theory

Many basic ‘divisibility’ properties are obvious yet tedious to formalize

$$\mathsf{gcd}(a, b) \neq 0 \Rightarrow \exists a'\ b'.a = a' \cdot \mathsf{gcd}(a, b) \wedge b = b' \cdot \mathsf{gcd}(a, b) \wedge \mathsf{coprime}(a', b')$$

Automation gap in formalizing complex analysis

$$|\|w - z\| - r| = d \wedge \|u - w\| < d/2 \wedge \|x - z\| = r \Rightarrow d/2 \leqslant \|x - u\|$$



This is not immediately solvable by HOL Light's standard automation, even though the analogous property over $\mathbb{R}$ would be.

Something positive at last . . .

These two examples have motivated the development of new decision procedures, and to further results in pure logic.