AAC conjecture for a simple and finer version of FLT

Shunichi Katoh

March 19, 2014

Abstract. This paper presents AAC conjecture for a simple and finer version of FLT (Fermat's Last Theorem) which had been FLC (Fermat's Last Conjecture) for more than 350 years since 1637 and was finally proved in the end of the 20th cencury in a profoundly sophisticated and complex way. However complex the proof is, FLC itself is very simple. It conjectures that a simple equation has no solutions. AAC conjecture comes from a pursuit of a simpler proof of FLC. It conjectures that two simple equations have no solutions except some evident ones. It is shown, in a rigorous and step-by-step way, that if AAC is true then FLC is true, and that AAC is a finer version of FLC. AAC conjecture will give us a finer view and an opportunity for finding a simple proof of FLC. If AAC conjecture is proved, without using FLT, then AAC theorem will be a theorem for itself, and FLT will be a beautiful specialization of AAC theorem.

1. Prefatory note

The numbers used are only integers (rational integers). Positive integers are used mainly. Arithmetic operands should be positive; the second operand of an exponentiation operation should not be negative.

2. Definitions of AAC conjecture and FLC

The domains of variables a,b,c,n,k are defined as follows throughout this paper.
a,b,c>0, n>2, k>=0
The following three equations are considered.
a^n+b^n=c^n   (equation f)
a^n+b^n=c^n*2^k (equation g)
a^n-b^n=c^n*2^k (equation h)
AAC conjecture (AAC) and Fermat's Last Conjecture (FLC) are defined as follows.
FLC states that the equation f has no solutions.
AAC states that the equations g and h have no solutions except the following ones of the equation g.
(a,b,c,n,k)=(a,a,c,n,1+j*n) (a=c*2^j, j>=0)
AAC is named after "a,a,c" in the above solutions.

3. Observation of the equations

The following hold about the above equations f, g and h.
g1: The equation g has following solutions where a=b.
(a,b,c,n,k)=(1,1,1,n,1) (included as c=1 in the following),
(a,b,c,n,k)=(c,c,c,n,1) (included as j=0 in the following),
(a,b,c,n,k)=(a,a,c,n,1+j*n) (a=c*2^j, j>=0)
f0: The equation f has no solutions where a=b.
g0: The equation g has no more solutions where a=b.
h0: The equation h has no solutions where a=b.
g1 is proved by substitution.
f0 and g0 are proved similarly as the square root of 2 cannot be a ratio of two natural numbers.
h0 is evident.
Therefore, AAC conjecture is true in its subdomain where a=b.
Below, AAC conjecture in its subdomain where a<>b and FLC in its domain are compared by their "binarized possibilities" of having solution(s), in a rigorous and step-by-step way, starting from a very basic standpoint.
Note: The comparison utilizes a homomorphism from the domain (a 4D number space) of FLC into the subdomain (a 5D number space) where a<>b of AAC conjecture in terms of "binarized possibility" for a point to be a solution.

4. Preparation for comparison of AAC conjecture and FLC

```
Notations are as follows.  Some are unique.
   "*" is the multiplication operator.  "*" is always written explicitly.
   "^" is the exponentiation operator.  x^k is defined, x^0=1 where x>0, k>=0.
   "/" is the division operator.
   "%" is the "bitwise inclusive or" operator when used between numbers.
   "%" is the "union" operator  when used between sets.
   "@" means "is a member of", not "which is a member of".
   "<" means "is a subset of"   when used between sets.
   ">" means "is a superset of" when used between sets.
   "^X" means "the powerset of a set X".    (not written as "2^X" in this paper)
   ":@" means "is defined as a member of".
   ":=" means "is defined as" or "is assigned the value of".
   "=:" means "defines" or "assigns the value to".
   ".:" means "therefore" or "then".       (as a right slanted "therefore" sign)
   ":." means "because" or "because of".      (as the oppsite sign of the above)
   "::" means "note:".
    :: The values of x and y are exchanged by (x,y):=(y,x).
Additional information is included in the appendix.
Let sets of numbers be defined as follows, including 4D and 5D number spaces.
   Z:={alle ganzen Zahlen} = {all               integers}={0,+-1,+-2,+-3,...}
   N:={all natural numbers}=:{all      positive integers}={  1,2,3,4,5,6,...}
   D:={all   odd   numbers}:={all  odd positive integers}={  1,  3,  5,  ...}
   E:={all  even   numbers}:={all even positive integers}={    2,  4,  6,...}
   K:={all counter numbers}:={all   nonnegative integers}={0,1,2,3,4,5,6,...}
   L:={all the numbers greater than 2}                   ={      3,4,5,6,...}
   2*K:={2*k; k@K}                                       ={0,  2,  4,  6,...}
   2^K:={2^k; k@K}                                       ={1,2,4,8,16,32,64,...}
   S:=(N,N,N,L)     (S is defined as a 4D number space.)
   T:=(N,N,N,L,K)   (T is defined as a 5D number space.)
   W:=(D,D,D,L,N)   (W is defined as a subspace of T.)
Let 4D,5D variables and some subspaces of W be defined as follows.
   an:=(a,b,c,n)    (an is defined as a variable for a point in S; not a*n.)
   ak:=(a,b,c,n,k)  (ak is defined as a variable for a point in T; not a*k.)
   xz:=(x,y,z)
   xn:=(x,y,z,n)
   xk:=(x,y,z,n,k)  (These parentheses are removed within other parentheses.)
   U:={(xn,k*n); xk@W, x>y}  (U is n-times coarser than V for each value of n.)
   V:={(xn, k ); xk@W, x>y}  (V is n-times  finer  than U for each value of n.)
     ={   xk   ; xk@W, x>y}  (V is about half of W (doobla ve).  U<V<W<T)
Let functions f(),g(),h() be defined as follows.
   f:S->Z,          f(xn):=f(x,y,z,n)  :=x^n+y^n-z^n
   g:T->Z, g(xn,k):=g(xk):=g(x,y,z,n,k):=x^n+y^n-z^n*2^k
   h:T->Z, h(xn,k):=h(xk):=h(x,y,z,n,k):=x^n-y^n-z^n*2^k
Let oddness d(), evenness e(), "evenness index" i(), "minimum evenness" em(),
"number of odd arguments" nda() be functions defined as follows.
   d:N->D,   d(x):=(greatest odd divisor of x)
   e:N->2^K, e(x):=x/d(x)
   i:N->K,   i(x):=(such number k that satisfies e(x)=2^k as well as x=d(x)*2^k)
   em:(N,N,...,N)->2^K, em(x,y,...,z):=min(e(x),e(y),...,e(z))
   nda:(N,N,...,N)->K, nda(x,y,...,z):=(number of odd arguments of nda())
Let "binarized possibility" bip() be a function defined as follows.
   P:={all the statements each of which is either true or false distinctly}
   q:@P   (q is defined as a variable for a member of P.)
   Q:@^P  (Q is defined as a variable for a subset of P.  Q<P as a result.)
   bip:P%^P->{1,0},               (:: 1 and 0 mean 100% and 0% respectively.)
   bip(q):=(if a statement q is true then be 1, else be 0),
   bip(Q):=bip(a statement set Q contains at least one true statement)
   :: bip(null set)=0, bip({q})=bip(q) by the above definition.
   :: If Q1>Q2 then bip(Q1)>=bip(Q2).  bip(Q1%Q2)=bip(Q1)%bip(Q2)  (Q1,Q2<P)
Let bip() have the following alternative forms, for flexibility.
   bip(q; q@Q):=bip({q; q@Q})=bip(Q)
   bip(equ; cnd):=bip(equation "equ" has solution(s) under condition "cnd")
```

5. Comparison of AAC conjecture and FLC

```
bip(equation f;            an@S)
 = bip(a^n+b^n=c^n;        an@S)
 = bip(f(an)=0;            an@S)
 = bip(f(an)/e^n=0;        an@S, e=em(a,b,c))
 = bip(f(a/e,b/e,c/e,n)=0; an@S, e=em(a,b,c))
 = bip(f(xn)=0;            xn@S, nda(x,y,z)>=1)     (:: xn:=(a/e,b/e,c/e,n))
 = bip(f(xn)=0;            xn@S, nda(x,y,z)>=1, nda(x^n,y^n,z^n)@2*K)
 = bip(f(xn)=0;            xn@S, nda(x,y,z)=2)
 = bip(f(xn)=0;            xn@S, xz@(D,D,E)%(D,E,D)%(E,D,D))
 = bip(f(xn)=0;            xn@S, xz@(D,D,E), x>=y)  (:. f(xn)=f(y,x,z,n))
  %bip(f(xn)=0;            xn@S, xz@(D,E,D))        (:. f(xn)=f(y,x,z,n))
 = bip(f(xn)=0;            xn@S, xz@(D,D,E), x>y)   (:. f(xn)<>0 if x=y.  :. f0)
  %bip(f(xn)=0;            xn@S, xz@(D,E,D), x<z)   (:. x^n+y^n=z^n)
 = bip(g(xn,0)=0                                    (:. g(xn,0)= f(xn))
    or h(xn,0)=0;          xn@(D,D,E,L), x>y)       (:. h(xn,0)=-f(y,z,x,n))
 = bip(g(xn,k*n)=0
    or h(xn,k*n)=0;        xk@(D,D,D,L,N), x>y)     (:: xk:=(x,y,d(z),n,i(z)))
 = bip(g(xn,k*n)=0 or h(xn,k*n)=0; xk@W, x>y)
 = bip(g(ak)=0 or h(ak)=0; ak@U)                    (:: ak:=(xn,k*n))
bip(equation g;            ak@T, a<>b)
 = bip(a^n+b^n=c^n*2^k;    ak@T, a<>b)
 = bip(g(ak)=0;            ak@T, a<>b)
 = bip(g(ak)/e=0;          ak@T, a<>b, e=em(a^n  ,b^n  ,c^n*2^k  ),
                                        nda(a^n/e,b^n/e,c^n*2^k/e)=2)
 = bip(g(xk)=0;            xk@W, x<>y)      (:: i(a)=i(b), i:=i(c)*n+k-i(a)*n>0,
                                                xk:=(d(a),d(b),d(c),n,i))
  %bip(h(xn,k*n)=0;        xk@W)            (:: i(a)*n=i(c)*n+k, i:=i(b)-i(a)>0,
                                                xk:=(d(c),d(a),d(b),n,i),
                                             :. g(ak)=g(b,a,c,n,k))
 = bip(g(xk)=0;            xk@W, x>y)       (:. g(xk)=g(y,x,z,n,k))
  %bip(h(xn,k*n)=0;        xk@W, x>y)       (:. x^n=y^n+z^n*2^(k*n))
 = bip(g(ak)=0;            ak@V)            (:: ak:=xk)
  %bip(h(ak)=0;            ak@U)            (:: ak:=(xn,k*n))
bip(equation h;            ak@T, a<>b)
 = bip(a^n-b^n=c^n*2^k;    ak@T, a<>b)
 = bip(h(ak)=0;            ak@T)            (:. h(ak)<>0 if a=b.)
 = bip(h(ak)/e=0;          ak@T, e=em(a^n  ,b^n  ,c^n*2^k  ),
                                  nda(a^n/e,b^n/e,c^n*2^k/e)=2)
 = bip(h(xk)=0;            xk@W)            (:: i(a)=i(b), i:=i(c)*n+k-i(a)*n>0,
                                                xk:=(d(a),d(b),d(c),n,i))
  %bip(h(xn,k*n)=0;        xk@W)            (:: i(a)*n=i(c)*n+k, i:=i(b)-i(a)>0,
                                                xk:=(d(a),d(c),d(b),n,i))
  %bip(g(xn,k*n)=0;        xk@W)            (:: i(b)*n=i(c)*n+k, i:=i(a)-i(b)>0,
                                                xk:=(d(b),d(c),d(a),n,i))
 = bip(h(xk)=0;            xk@W, x>y)       (:. x^n=y^n+z^n*2^k)
  %bip(h(xn,k*n)=0;        xk@W, x>y)       (:. x^n=y^n+z^n*2^(k*n))
  %bip(g(xn,k*n)=0;        xk@W, x>y)       (:. g(xn,k*n)=g(y,x,z,n,k*n) and
                                             :. g(xn,k*n)<>0 if x=y.  :. g0)
 = bip(h(ak)=0;            ak@V)            (:. ak:=xk)
  %bip(h(ak)=0;            ak@U)            (:: ak:=(xn,k*n))
  %bip(g(ak)=0;            ak@U)            (:: ak:=(xn,k*n))
 = bip(h(ak)=0;            ak@V)            (:. V>U)
  %bip(g(ak)=0;            ak@U)
.: bip(equation g     ; ak@T, a<>b) = bip(g(ak)=0; ak@V) % bip(h(ak)=0; ak@U)
   bip(equation      h; ak@T, a<>b) = bip(g(ak)=0; ak@U) % bip(h(ak)=0; ak@V)
.: bip(equation g or h; ak@T, a<>b) = bip(g(ak)=0 or h(ak)=0; ak@V)  (:. V>U)
   bip(equation  f    ; an@S      ) = bip(g(ak)=0 or h(ak)=0; ak@U)
.: bip(equation g or h; ak@T, a<>b)>= bip(equation f; an@S)          (:. V>U)
```

If AAC conjecture is true, then leftside=0, then rightside=0, then FLC is true.
It is proved thus that if AAC conjecture is true then FLC is true.
It is proved also that AAC conjecture is a finer version of FLC.
(:. V is n-times finer than U for each value of n.)

6. Expectation

As a finer version of FLC, AAC conjecture will give us a finer view of FLC, help us understand FLC better, and provide us with an opportunity for finding a simple proof of both itself and FLC at the same time. If AAC conjecture is proved, without using FLT, then AAC theorem will be a theorem for itself, and FLT will be a beautiful specialization of AAC theorem.

Appendix

Note 1: Characters used

This paper is written using only ASCII characters. Neither superscripts nor subscripts are used. Some notations may be unique as follows. (cf. Page 2 of 4)
"%" means "or" depending on the context, because "%" resembles "or" when it is written by hand. It is used as the "bitwise inclusive or" operator between numbers, and as the "union" operator between sets.
"@" means "is a member of", not "which is a member of", as a mutatis mutandis application of "@" in an E-mail address.
This style and the naming convention have been gradually developed along with mathematical contents in this research. Most work including thinking has been done on a handy off-line PC with a compact but versatile text-editor, whose combination has been carried and used conveniently almost anytime and anywhere.

Note 2: Naming convention

Names are used for variables, named constants, functions, equations, statements, conditions, in either member or set level. These are named in ASCII characters.
Each name is an alphabet optionally followed by alphabet(s) and/or digit(s).
A name referring to a set begins with a capital letter, and vice versa.
The name of a function is immediately succeeded by parentheses, and vice versa.
The succeeding parentheses is a part of the identifier of the function.
The leftmost y below does not mean a variable y and could be written as y().

```
   y:Dom->Ran; y(x):=(such and such)
```

A function and an identically named object can be defined either dependently or independently. A variable or a named constant cannot be differentiated by the name. For example, following names are used in this paper.

```
   a,b,c,n,k,e,i,an,ak are variable names.
   f(),g(),h(),i(),em(),bip() are function identifiers.
   f,g,h are equation names.
   f0,g0,h0,g1 are constant statement names.
   q is a variable statement name.
   Q is a variable statement set name.
   Z,N,D,E,S,T are constant set names.
```

Note 3: Definition of "counter number"

Natural numbers come from counting. At the instant we count "n", the internal state of a real or virtual counter changes from n-1 to n during count-up, and from n to n-1 during count-down. (n>0) Therefore the domain of "counter number" is downward wider by 1 than that of "counting number", that is, "natural number" which defines "positive integer". Therefore, in this paper, "counting number", "natural number" and "positive integer" are considered synonymous, and naturally "counter number" is defined as "nonnegative integer".

Shunichi Katoh (KATOH ShunIchi)
E-mail address: katoh.shun.ichi@nifty.com

The family name Katoh is pronounced like *ka* in kaiser and *to* in token; the first part Shu of the personal name is like *shou* in should, and when the succeeding n is pronounced, breath goes out through not only nose but also mouth, and it is not liaisoned to its succeeding sound i. (Just for information.)