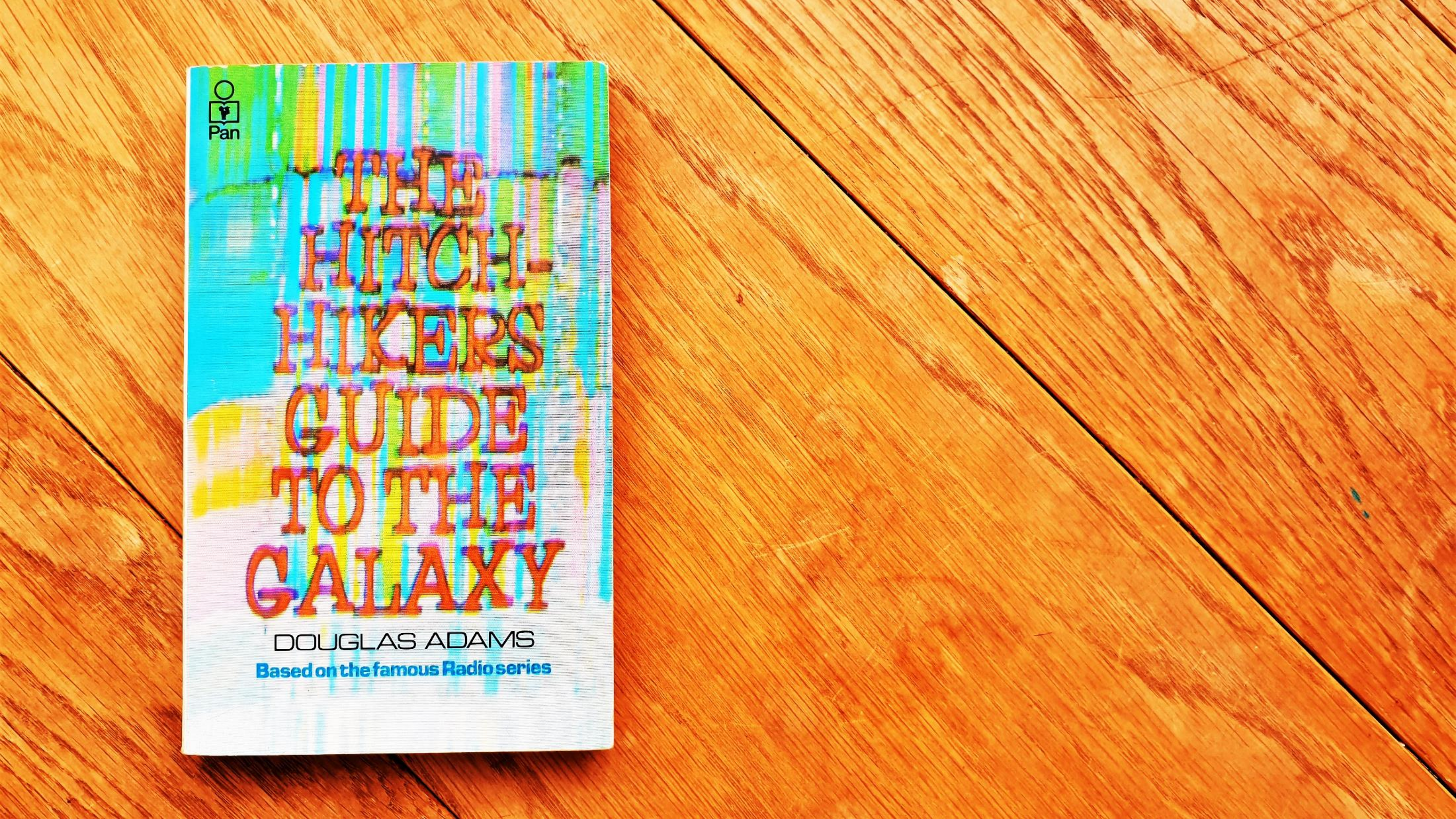
Code as Life

@KevlinHenney

42
42



Pan

THE HITCH-HIKERS GUIDE TO THE GALAXY

DOUGLAS ADAMS

Based on the famous Radio series

Pan
THE HITCH HIKERS GUIDE TO THE GALAXY
DOUGLAS ADAMS
Based on the famous Radio series

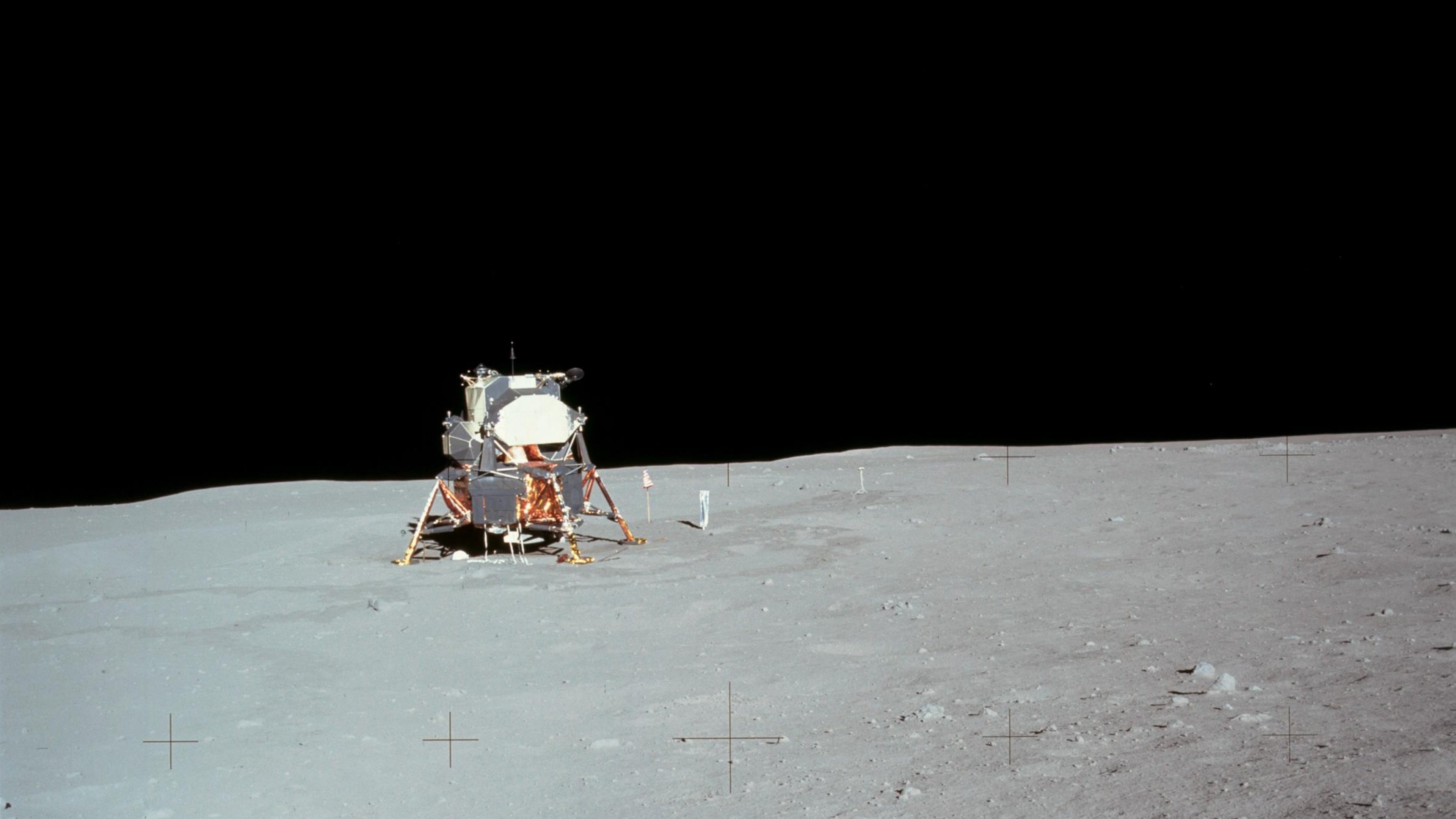
We demand rigidly defined areas of doubt and uncertainty!

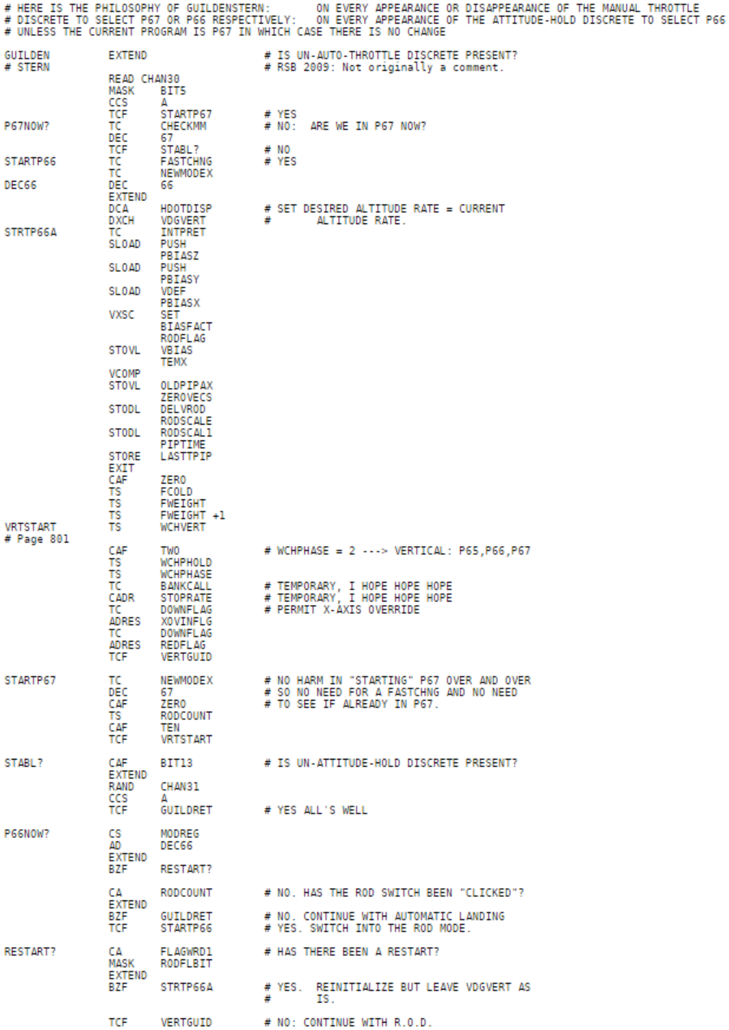
code

code, *noun*

- a set of instructions for a computer

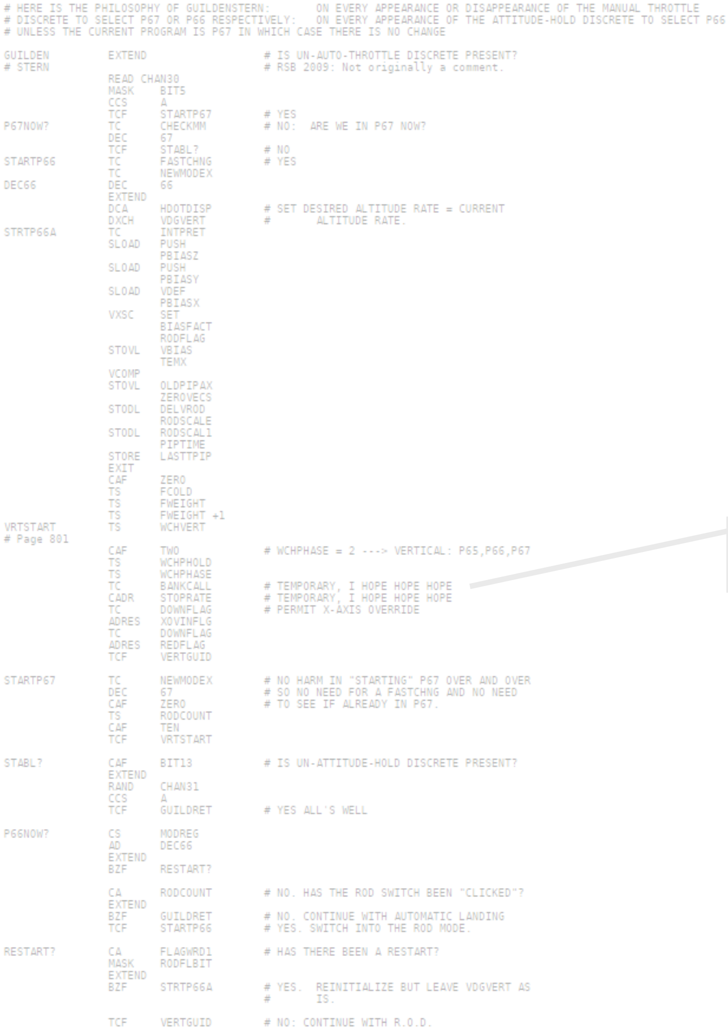
Concise Oxford English Dictionary · Oxford English Dictionary · Merriam-Webster’s Collegiate Dictionary





```
# HERE IS THE PHILOSOPHY OF GUILDENSTERN:       ON EVERY APPEARANCE OR DISAPPEARANCE OF THE MANUAL THROTTLE
# DISCRETE TO SELECT P67 OR P66 RESPECTIVELY:   ON EVERY APPEARANCE OF THE ATTITUDE-HOLD DISCRETE TO SELECT P66
# UNLESS THE CURRENT PROGRAM IS P67 IN WHICH CASE THERE IS NO CHANGE

GUILDEN         EXTEND                  # IS UN-AUTO-THROTTLE DISCRETE PRESENT?
# STERN                                 # RSB 2009: Not originally a comment.
                READ CHAN30
                MASK    BIT5
                CCS     A
                TCF     STARTP67        # YES
P67NOW?         TC      CHECKMM         # NO:  ARE WE IN P67 NOW?
                DEC     67
                TCF     STABL?          # NO
STARTP66        TC      FASTCHNG        # YES
                TC      NEWMODEX
DEC66           DEC     66
                EXTEND
                DCA     HDOTDISP        # SET DESIRED ALTITUDE RATE = CURRENT
                DXCH    VDGVERT         #       ALTITUDE RATE.
STRTP66A        TC      INTPRET
                SLOAD   PUSH
                        PBIASZ
                SLOAD   PUSH
                        PBIASY
                SLOAD   VDEF
                        PBIASX
                VXSC    SET
                        BIASFACT
                        RODFLAG
                STOVL   VBIAS
                        TEMX
                VCOMP
                STOVL   OLDPIPAX
                        ZEROVECS
                STODL   DELVROD
                        RODSCALE
                STODL   RODSCAL1
                        PIPTIME
                STORE   LASTTPIP
                EXIT
                CAF     ZERO
                TS      FCOLD
                TS      FWEIGHT
                TS      FWEIGHT +1
VRTSTART        TS      WCHVERT
# Page 801
                CAF     TWO             # WCHPHASE = 2 ---> VERTICAL: P65,P66,P67
                TS      WCHPHOLD
                TS      WCHPHASE
                TC      BANKCALL        # TEMPORARY, I HOPE HOPE HOPE
                CADR    STOPRATE        # TEMPORARY, I HOPE HOPE HOPE
                TC      DOWNFLAG        # PERMIT X-AXIS OVERRIDE
                ADRES   XOVINFLG
                TC      DOWNFLAG
                ADRES   REDFLAG
                TCF     VERTGUID

STARTP67        TC      NEWMODEX        # NO HARM IN "STARTING" P67 OVER AND OVER
                DEC     67              # SO NO NEED FOR A FASTCHNG AND NO NEED
                CAF     ZERO            # TO SEE IF ALREADY IN P67.
                TS      RODCOUNT
                CAF     TEN
                TCF     VRTSTART

STABL?          CAF     BIT13           # IS UN-ATTITUDE-HOLD DISCRETE PRESENT?
                EXTEND
                RAND    CHAN31
                CCS     A
                TCF     GUILDRET        # YES ALL'S WELL

P66NOW?         CS      MODREG
                AD      DEC66
                EXTEND
                BZF     RESTART?

                CA      RODCOUNT        # NO. HAS THE ROD SWITCH BEEN "CLICKED"?
                EXTEND
                BZF     GUILDRET        # NO. CONTINUE WITH AUTOMATIC LANDING
                TCF     STARTP66        # YES. SWITCH INTO THE ROD MODE.

RESTART?        CA      FLAGWRD1        # HAS THERE BEEN A RESTART?
                MASK    RODFLBIT
                EXTEND
                BZF     STRTP66A        # YES.  REINITIALIZE BUT LEAVE VDGVERT AS
                                        #       IS.

                TCF     VERTGUID        # NO: CONTINUE WITH R.O.D.
```

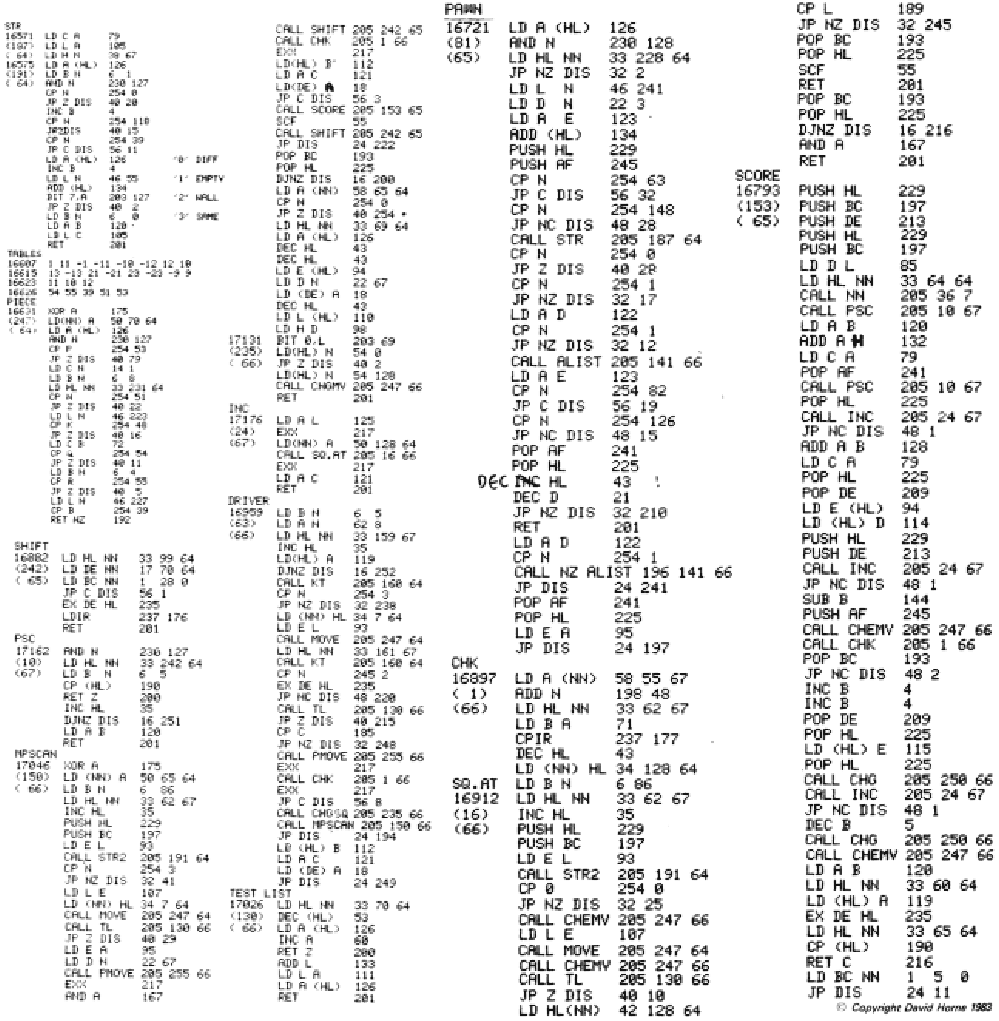
http://lambda-the-ultimate.org/node/3522



```
# HERE IS THE PHILOSOPHY OF GUILDENSTERN:       ON EVERY APPEARANCE OR DISAPPEARANCE OF THE MANUAL THROTTLE
# DISCRETE TO SELECT P67 OR P66 RESPECTIVELY:   ON EVERY APPEARANCE OF THE ATTITUDE-HOLD DISCRETE TO SELECT P66
# UNLESS THE CURRENT PROGRAM IS P67 IN WHICH CASE THERE IS NO CHANGE

GUILDEN         EXTEND                  # IS UN-AUTO-THROTTLE DISCRETE PRESENT?
# STERN                                 # RSB 2009: Not originally a comment.
                READ CHAN30
                MASK    BIT5
                CCS     A
                TCF     STARTP67        # YES
P67NOW?         TC      CHECKMM         # NO:  ARE WE IN P67 NOW?
                DEC     67
                TCF     STABL?          # NO
STARTP66        TC      FASTCHNG        # YES
                TC      NEWMODEX
DEC66           DEC     66
                EXTEND
                DCA     HDOTDISP        # SET DESIRED ALTITUDE RATE = CURRENT
                DXCH    VDGVERT         #       ALTITUDE RATE.
STRTP66A        TC      INTPRET
                SLOAD   PUSH
                        PBIASZ
                SLOAD   PUSH
                        PBIASY
                SLOAD   VDEF
                        PBIASX
                VXSC    SET
                        BIASFACT
                        RODFLAG
                STOVL   VBIAS
                        TEMX
                VCOMP
                STOVL   OLDPIPAX
                        ZEROVECS
                STODL   DELVROD
                        RODSCALE
                STODL   RODSCAL1
                        PIPTIME
                STORE   LASTTPIP
                EXIT
                CAF     ZERO
                TS      FCOLD
                TS      FWEIGHT
                TS      FWEIGHT +1
VRTSTART        TS      WCHVERT
# Page 801
                CAF     TWO             # WCHPHASE = 2 ---> VERTICAL: P65,P66,P67
                TS      WCHPHOLD
                TS      WCHPHASE
                TC      BANKCALL        # TEMPORARY, I HOPE HOPE HOPE
                CADR    STOPRATE        # TEMPORARY, I HOPE HOPE HOPE
                TC      DOWNFLAG        # PERMIT X-AXIS OVERRIDE
                ADRES   XOVINFLG
                TC      DOWNFLAG
                ADRES   REDFLAG
                TCF     VERTGUID

STARTP67        TC      NEWMODEX        # NO HARM IN "STARTING" P67 OVER AND OVER
                DEC     67              # SO NO NEED FOR A FASTCHNG AND NO NEED
                CAF     ZERO            # TO SEE IF ALREADY IN P67.
                TS      RODCOUNT
                CAF     TEN
                TCF     VRTSTART

STABL?          CAF     BIT13           # IS UN-ATTITUDE-HOLD DISCRETE PRESENT?
                EXTEND
                RAND    CHAN31
                CCS     A
                TCF     GUILDRET        # YES ALL'S WELL

P66NOW?         CS      MODREG
                AD      DEC66
                EXTEND
                BZF     RESTART?

                CA      RODCOUNT        # NO. HAS THE ROD SWITCH BEEN "CLICKED"?
                EXTEND
                BZF     GUILDRET        # NO. CONTINUE WITH AUTOMATIC LANDING
                TCF     STARTP66        # YES. SWITCH INTO THE ROD MODE.

RESTART?        CA      FLAGWRD1        # HAS THERE BEEN A RESTART?
                MASK    RODFLBIT
                EXTEND
                BZF     STRTP66A        # YES.  REINITIALIZE BUT LEAVE VDGVERT AS
                                        #       IS.

                TCF     VERTGUID        # NO: CONTINUE WITH R.O.D.
```

TEMPORARY, I HOPE HOPE HOPE
TEMPORARY, I HOPE HOPE HOPE

http://lambda-the-ultimate.org/node/3522



```
STR
16571  LD C A       79
(197)  LD L A       105
( 64)  LD H N       38 67
16575  LD A (HL)    126
(191)  LD B N       6  1
( 64)  AND N        230 127
       CP N         254 0
       JP Z DIS     40 20
       INC B        4
       CP N         254 110
       JPZDIS       40 15
       CP N         254 39
       JP C DIS     56 11
       LD A (HL)    126          '0' DIFF
       INC B        4
       LD L N       46 55        '1' EMPTY
       ADD (HL)     134
       BIT 7.A      203 127      '2' WALL
       JP Z DIS     40  2
       LD B N       6   0        '3' SAME
       LD A B       120
       LD L C       105
       RET          201
TABLES
16607  1 11 -1 -11 -10 -12 12 10
16615  13 -13 21 -21 23 -23 -9 9
16623  11 10 12
16626  54 55 39 51 53
PIECE
16631  XOR A        175
(247)  LD(NN) A     50 70 64
( 64)  LD A (HL)    126
       AND N        230 127
       CP P         254 53
       JP Z DIS     40 79
       LD C N       14 1
       LD B N       6  8
       LD HL NN     33 231 64
       CP N         254 51
       JP Z DIS     40 22
       LD L N       46 223
       CP K         254 48
       JP Z DIS     40 16
       LD C B       72
       CP Q         254 54
       JP Z DIS     40 11
       LD B N       6  4
       CP R         254 55
       JP Z DIS     40  5
       LD L N       46 227
       CP B         254 39
       RET NZ       192

SHIFT
16882  LD HL NN     33 99 64
(242)  LD DE NN     17 70 64
( 65)  LD BC NN     1  28 0
       JP C DIS     56 1
       EX DE HL     235
       LDIR         237 176
       RET          201
PSC
17162  AND N        230 127
(10)   LD HL NN     33 242 64
(67)   LD B  N      6  5
       CP (HL)      190
       RET Z        200
       INC HL       35
       DJNZ DIS     16 251
       LD A B       120
       RET          201
MPSCAN
17046  XOR A        175
(150)  LD (NN) A    50 65 64
( 66)  LD B N       6  86
       LD HL NN     33 62 67
       INC HL       35
       PUSH HL      229
       PUSH BC      197
       LD E L       93
       CALL STR2    205 191 64
       CP N         254 3
       JP NZ DIS    32 41
       LD L E       107
       LD (NN) HL   34 7 64
       CALL MOVE    205 247 64
       CALL TL      205 130 66
       JP Z DIS     40 29
       LD E A       95
       LD D N       22 67
       CALL PMOVE   205 255 66
       EXX          217
       AND A        167
```

```
       CALL SHIFT   205 242 65
       CALL CHK     205 1 66
       EXX          217
       LD(HL) B     112
       LD A C       121
       LD(DE) A     18
       JP C DIS     56 3
       CALL SCORE   205 153 65
       SCF          55
       CALL SHIFT   205 242 65
       JP DIS       24 222
       POP BC       193
       POP HL       225
       DJNZ DIS     16 200
       LD A (NN)    58 65 64
       CP N         254 0
       JP Z DIS     40 254 •
       LD HL NN     33 69 64
       LD A (HL)    126
       DEC HL       43
       DEC HL       43
       LD E (HL)    94
       LD D N       22 67
       LD (DE) A    18
       DEC HL       43
       LD L (HL)    110
       LD H D       98
17131  BIT 0.L      203 69
(235)  LD(HL) N     54 0
( 66)  JP Z DIS     40 2
       LD(HL) N     54 128
       CALL CHGMV   205 247 66
       RET          201
INC
17176  LD A L       125
(24)   EXX          217
(67)   LD(NN) A     50 128 64
       CALL SQ.AT   205 16 66
       EXX          217
       LD A C       121
       RET          201
DRIVER
16959  LD B N       6  5
(63)   LD A N       62 8
(66)   LD HL NN     33 159 67
       INC HL       35
       LD(HL) A     119
       DJNZ DIS     16 252
       CALL KT      205 160 64
       CP N         254 3
       JP NZ DIS    32 238
       LD (NN) HL   34 7 64
       LD E L       93
       CALL MOVE    205 247 64
       LD HL NN     33 161 67
       CALL KT      205 160 64
       CP N         245 2
       EX DE HL     235
       JP NC DIS    48 220
       CALL TL      205 130 66
       JP Z DIS     40 215
       CP C         185
       JP NZ DIS    32 248
       CALL PMOVE   205 255 66
       EXX          217
       CALL CHK     205 1 66
       EXX          217
       JP C DIS     56 8
       CALL CHGSQ   205 235 66
       CALL MPSCAN  205 150 66
       JP DIS       24 194
       LD (HL) B    112
       LD A C       121
       LD (DE) A    18
       JP DIS       24 249
TEST LIST
17026  LD HL NN     33 70 64
(130)  DEC (HL)     53
( 66)  LD A (HL)    126
       INC A        60
       RET Z        200
       ADD L        133
       LD L A       111
       LD A (HL)    126
       RET          201
```

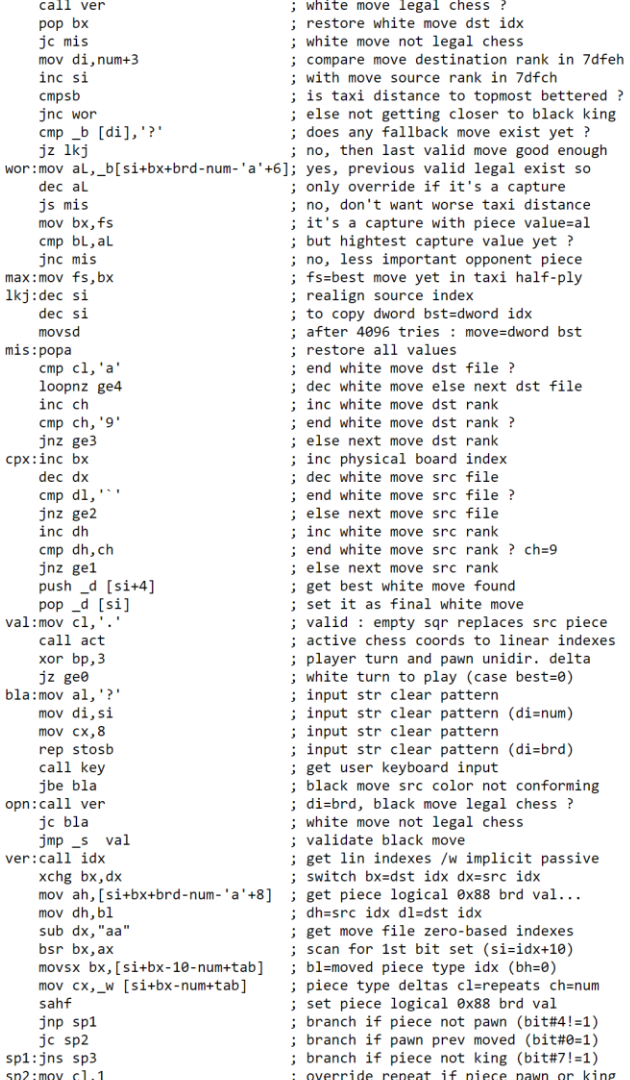
```
PAWN
16721  LD A (HL)    126
(81)   AND N        230 128
(65)   LD HL NN     33 228 64
       JP NZ DIS    32 2
       LD L  N      46 241
       LD D  N      22 3
       LD A  E      123
       ADD (HL)     134
       PUSH HL      229
       PUSH AF      245
       CP N         254 63
       JP C DIS     56 32
       CP N         254 148
       JP NC DIS    48 28
       CALL STR     205 187 64
       CP N         254 0
       JP Z DIS     40 28
       CP N         254 1
       JP NZ DIS    32 17
       LD A D       122
       CP N         254 1
       JP NZ DIS    32 12
       CALL ALIST   205 141 66
       LD A E       123
       CP N         254 82
       JP C DIS     56 19
       CP N         254 126
       JP NC DIS    48 15
       POP AF       241
       POP HL       225
   DEC ~~INC~~ HL       43   !
       DEC D        21
       JP NZ DIS    32 210
       RET          201
       LD A D       122
       CP N         254 1
       CALL NZ ALIST 196 141 66
       JP DIS       24 241
       POP AF       241
       POP HL       225
       LD E A       95
       JP DIS       24 197

CHK
16897  LD A (NN)    58 55 67
( 1)   ADD N        198 48
(66)   LD HL NN     33 62 67
       LD B A       71
       CPIR         237 177
       DEC HL       43
       LD (NN) HL   34 128 64
SQ.AT
16912  LD B N       6 86
(16)   LD HL NN     33 62 67
(66)   INC HL       35
       PUSH HL      229
       PUSH BC      197
       LD E L       93
       CALL STR2    205 191 64
       CP 0         254 0
       JP NZ DIS    32 25
       CALL CHEMV   205 247 66
       LD L E       107
       CALL MOVE    205 247 64
       CALL CHEMV   205 247 66
       CALL TL      205 130 66
       JP Z DIS     40 10
       LD HL(NN)    42 128 64
```

```
       CP L         189
       JP NZ DIS    32 245
       POP BC       193
       POP HL       225
       SCF          55
       RET          201
       POP BC       193
       POP HL       225
       DJNZ DIS     16 216
       AND A        167
       RET          201
SCORE
16793  PUSH HL      229
(153)  PUSH BC      197
( 65)  PUSH DE      213
       PUSH HL      229
       PUSH BC      197
       LD D L       85
       LD HL NN     33 64 64
       CALL NN      205 36 7
       CALL PSC     205 10 67
       LD A B       120
       ADD A H      132
       LD C A       79
       POP AF       241
       CALL PSC     205 10 67
       POP HL       225
       CALL INC     205 24 67
       JP NC DIS    48 1
       ADD A B      128
       LD C A       79
       POP HL       225
       POP DE       209
       LD E (HL)    94
       LD (HL) D    114
       PUSH HL      229
       PUSH DE      213
       CALL INC     205 24 67
       JP NC DIS    48 1
       SUB B        144
       PUSH AF      245
       CALL CHEMV   205 247 66
       CALL CHK     205 1 66
       POP BC       193
       JP NC DIS    48 2
       INC B        4
       INC B        4
       POP DE       209
       POP HL       225
       LD (HL) E    115
       POP HL       225
       CALL CHG     205 250 66
       CALL INC     205 24 67
       JP NC DIS    48 1
       DEC B        5
       CALL CHG     205 250 66
       CALL CHEMV   205 247 66
       LD A B       120
       LD HL NN     33 60 64
       LD (HL) A    119
       EX DE HL     235
       LD HL NN     33 65 64
       CP (HL)      190
       RET C        216
       LD BC NN     1  5  0
       JP DIS       24 11
```

code, *noun*

- a set of instructions for a computer
- a computer program, or a portion thereof

Concise Oxford English Dictionary · Oxford English Dictionary · Merriam-Webster’s Collegiate Dictionary



```
    call ver                      ; white move legal chess ?
    pop bx                        ; restore white move dst idx
    jc mis                        ; white move not legal chess
    mov di,num+3                  ; compare move destination rank in 7dfeh
    inc si                        ; with move source rank in 7dfch
    cmpsb                         ; is taxi distance to topmost bettered ?
    jnc wor                       ; else not getting closer to black king
    cmp _b [di],'?'               ; does any fallback move exist yet ?
    jz lkj                        ; no, then last valid move good enough
wor:mov aL,_b[si+bx+brd-num-'a'+6]; yes, previous valid legal exist so
    dec aL                        ; only override if it's a capture
    js mis                        ; no, don't want worse taxi distance
    mov bx,fs                     ; it's a capture with piece value=al
    cmp bL,aL                     ; but hightest capture value yet ?
    jnc mis                       ; no, less important opponent piece
max:mov fs,bx                     ; fs=best move yet in taxi half-ply
lkj:dec si                        ; realign source index
    dec si                        ; to copy dword bst=dword idx
    movsd                         ; after 4096 tries : move=dword bst
mis:popa                          ; restore all values
    cmp cl,'a'                    ; end white move dst file ?
    loopnz ge4                    ; dec white move else next dst file
    inc ch                        ; inc white move dst rank
    cmp ch,'9'                    ; end white move dst rank ?
    jnz ge3                       ; else next move dst rank
cpx:inc bx                        ; inc physical board index
    dec dx                        ; dec white move src file
    cmp dl,'`'                    ; end white move src file ?
    jnz ge2                       ; else next move src file
    inc dh                        ; inc white move src rank
    cmp dh,ch                     ; end white move src rank ? ch=9
    jnz ge1                       ; else next move src rank
    push _d [si+4]                ; get best white move found
    pop _d [si]                   ; set it as final white move
val:mov cl,'.'                    ; valid : empty sqr replaces src piece
    call act                      ; active chess coords to linear indexes
    xor bp,3                      ; player turn and pawn unidir. delta
    jz ge0                        ; white turn to play (case best=0)
bla:mov al,'?'                    ; input str clear pattern
    mov di,si                     ; input str clear pattern (di=num)
    mov cx,8                      ; input str clear pattern
    rep stosb                     ; input str clear pattern (di=brd)
    call key                      ; get user keyboard input
    jbe bla                       ; black move src color not conforming
opn:call ver                      ; di=brd, black move legal chess ?
    jc bla                        ; white move not legal chess
    jmp _s  val                   ; validate black move
ver:call idx                      ; get lin indexes /w implicit passive
    xchg bx,dx                    ; switch bx=dst idx dx=src idx
    mov ah,[si+bx+brd-num-'a'+8]  ; get piece logical 0x88 brd val...
    mov dh,bl                     ; dh=src idx dl=dst idx
    sub dx,"aa"                   ; get move file zero-based indexes
    bsr bx,ax                     ; scan for 1st bit set (si=idx+10)
    movsx bx,[si+bx-10-num+tab]   ; bl=moved piece type idx (bh=0)
    mov cx,_w [si+bx-num+tab]     ; piece type deltas cl=repeats ch=num
    sahf                          ; set piece logical 0x88 brd val
    jnp sp1                       ; branch if piece not pawn (bit#4!=1)
    jc sp2                        ; branch if pawn prev moved (bit#0=1)
sp1:jns sp3                       ; branch if piece not king (bit#7!=1)
sp2:mov cl,1                      ; override repeat if piece pawn or king
```

http://olivier.poudade.free.fr/src/BootChess.asm

```
public static int binarySearch(int[] a, int key) {
     int low = 0;
     int high = a.length - 1;

     while (low <= high) {
          int mid = (low + high) / 2;
          int midVal = a[mid];

          if (midVal < key)
               low = mid + 1
          else if (midVal > key)
               high = mid - 1;
          else
               return mid; // key found
     }
     return -(low + 1);  // key not found.
}
```

https://ai.googleblog.com/2006/06/extra-extra-read-all-about-it-nearly.html

code, *noun*

- a set of instructions for a computer
- a computer program, or a portion thereof
- a system of words, figures or symbols used to represent others

Concise Oxford English Dictionary · Oxford English Dictionary · Merriam-Webster's Collegiate Dictionary

Basic Date ___
Changed ___

LM-3

00611 No TIG For Given Elevation Angle

00777 PIPA fail caused the ISS Warning

01102 LGC Self Test Error

01103 *Unused CCS Branch Executed
Perform LGC Self Test

01104 *Delay Routine Busy
Perform LGC SELF TEST

01105 Downlink Too Fast
RSET If Alarm Recurrs, DOWNLINK FAILURE

01106 Uplink Too Fast
RSET If Alarm Recurs, UPLINK FAILURE

01107 Phase Table Failure
Perform 1. V74 LGC DOWNLINK
2. P27 As Necessary
3. V48 As Necessary
4. Reestablish REFSMMAT via P51
If FRESH START Recurs, LGC FAILURE

01201 *Executive Overflow - No Vac. Areas
*Perform LGC SELF TEST
RESET, then Recall EXTENDED VERB IF IT IS NOT FIXED

01202 *Executive Overflow - No Core Sets
Perform LGC SELF TEST
RESET, THEN RECALL EXTENDED VERB IF IT IS NOT FIXED.

code, *noun*

- a set of instructions for a computer
- a computer program, or a portion thereof
- a system of words, figures or symbols used to represent others, especially for the purposes of secrecy

Concise Oxford English Dictionary · Oxford English Dictionary · Merriam-Webster’s Collegiate Dictionary

```
48565078965739782930984189469428613770744208735135792401965207366869851
34010472374469687974399261175109737777010274475280490588313840375497099
87909653955227011712157025974666993240226834596619606034851742497735846
85188556745702571254749996482194184655710084119086259716947970799152004
86670997592359606132072597379799361886063169144735883002453369727818139
14797955513399949394882899846917836100182597890103160196183503434489568
70538452085380458424156548248893338047475871128339598968522325446084089
71119771276941207958624405471613210050064598201769617718094781136220027
23448272249323259547234688002927776497906148129840428345720146348968547
16908235473783566197218622496943162271666393905543024156473292485524899
12257394665486271404821171381243882177176029841255244647445055834628144
88335631902725319590439283873764073916891257924055015620889787163375999
10788708490815909754801928576845198859630532382349055809203299960323447
11407760198471635311617130785760848622363702835701049612595681846785965
33310077017991614674472549272833486916000647585917462781212690073518309
24153010630289329566584366200080047677896798438209079761985949364630938
05863367214696959750279687712057249966669805614533820741203159337703099
49152746918356593762102220068126798273445760938020304479122774980917955
93838712100058876668925844870047077255249706044465212713040432118261010
3591186476662963858495087448497373476861420880529443
```

```
                                                                        char
                                                            _3141592654[3141
          ],__3141[3141];_314159[31415],_3141[31415];main(){register char*
      _3_141,*_3_1415, *_3__1415; register int _314,_31415,__31415,*_31,
    _3_14159,__3_1415;*_3141592654=__31415=2,_3141592654[0][_3141592654
   -1]=1[__3141]=5;__3_1415=1;do{_3_14159=_314=0,__31415++;for( _31415
  =0;_31415<(3,14-4)*__31415;_31415++)_31415[_3141]=_314159[_31415]= -
1;_3141[*_314159=_3_14159]=_314;_3_141=_3141592654+__3_1415;_3_1415=
__3_1415    +__3141;for                        (_31415 = 3141-
           __3_1415  ;                        _31415;_31415--
           ,_3_141 ++,                        _3_1415++){_314
           +=_314<<2 ;                        _314<<=1;_314+=
          *_3_1415;_31                         =_314159+_314;
          if(!(*_31+1)                         )* _31 =_314 /
          __31415,_314                         [_3141]=_314 %
          __31415 ;* (                         _3__1415=_3_141
         )+= *_3_1415                          = *_31;while(*
         _3__1415 >=                          31415/3141 ) *
         _3__1415+= -                          10,(*--_3__1415
        )++;_314=_314                          [_3141]; if ( !
        _3_14159 && *                          _3_1415)_3_14159
        =1,__3_1415 =                          3141-_31415;}if(
        _314+(__31415                           >>1)>=__31415 )
        while ( ++ *                           _3_141==3141/314
       )*_3_141--=0                           ;}while(_3_14159
       ) ; { char *                           __3_14= "3.1415";
       write((3,1),                           (--*__3_14,__3_14
       ),(_3_14159                            ++,++_3_14159))+
      3.1415926; }                            for ( _31415 = 1;
     _31415<3141-                            1;_31415++)write(
    31415% 314-(                            3,14),_3141592654[
  _31415    ] +                            "0123456789","314"
  [ 3]+1)-_314;                            puts((*_3141592654=0
,_3141592654))                             ;_314= *"3.141592";}
```

code, *noun*

- a set of instructions for a computer
- a computer program, or a portion thereof
- a system of words, figures or symbols used to represent others, especially for the purposes of secrecy
- a set of conventions or principles governing behaviour or activity in a particular domain

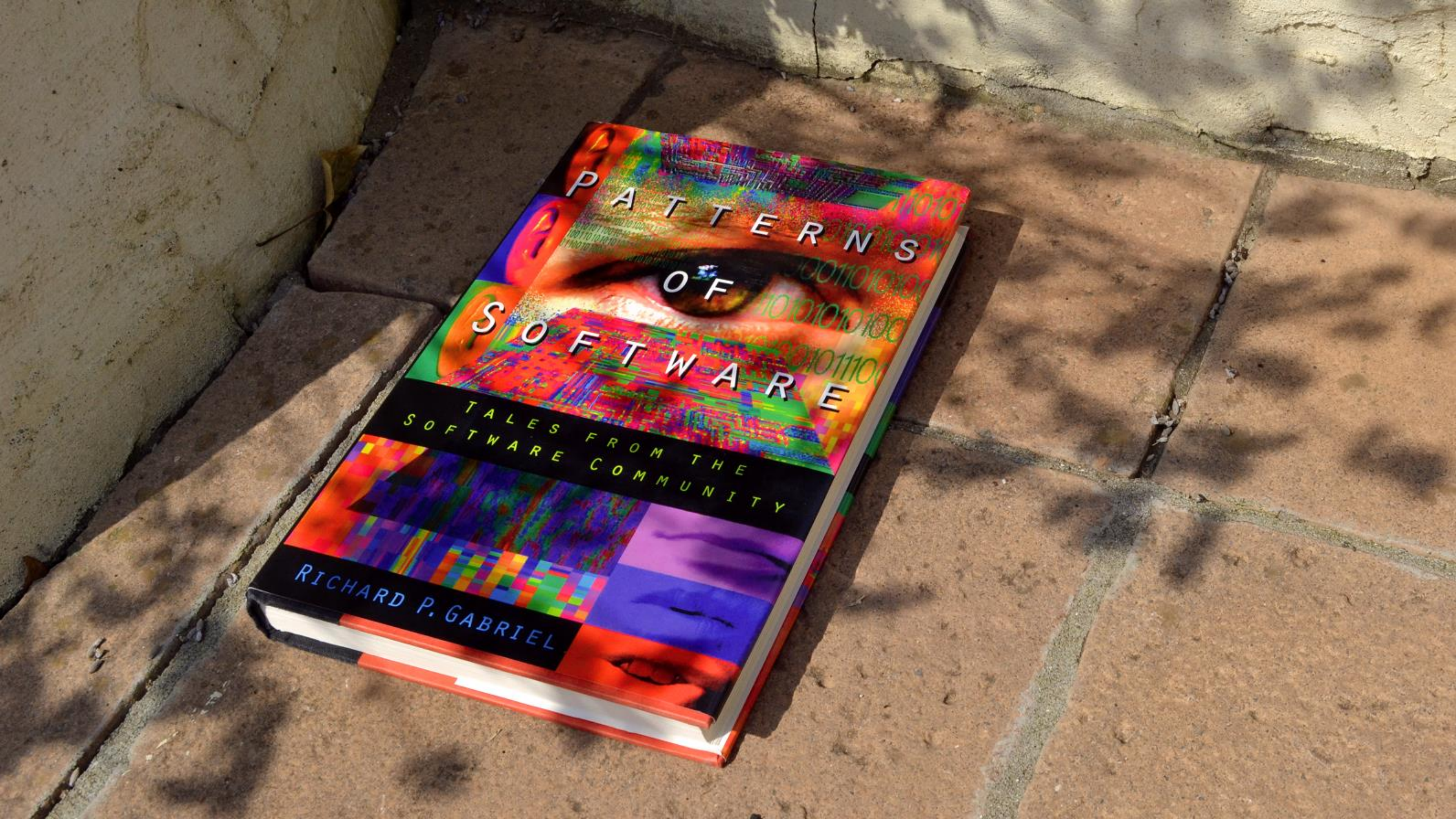
Concise Oxford English Dictionary · Oxford English Dictionary · Merriam-Webster's Collegiate Dictionary

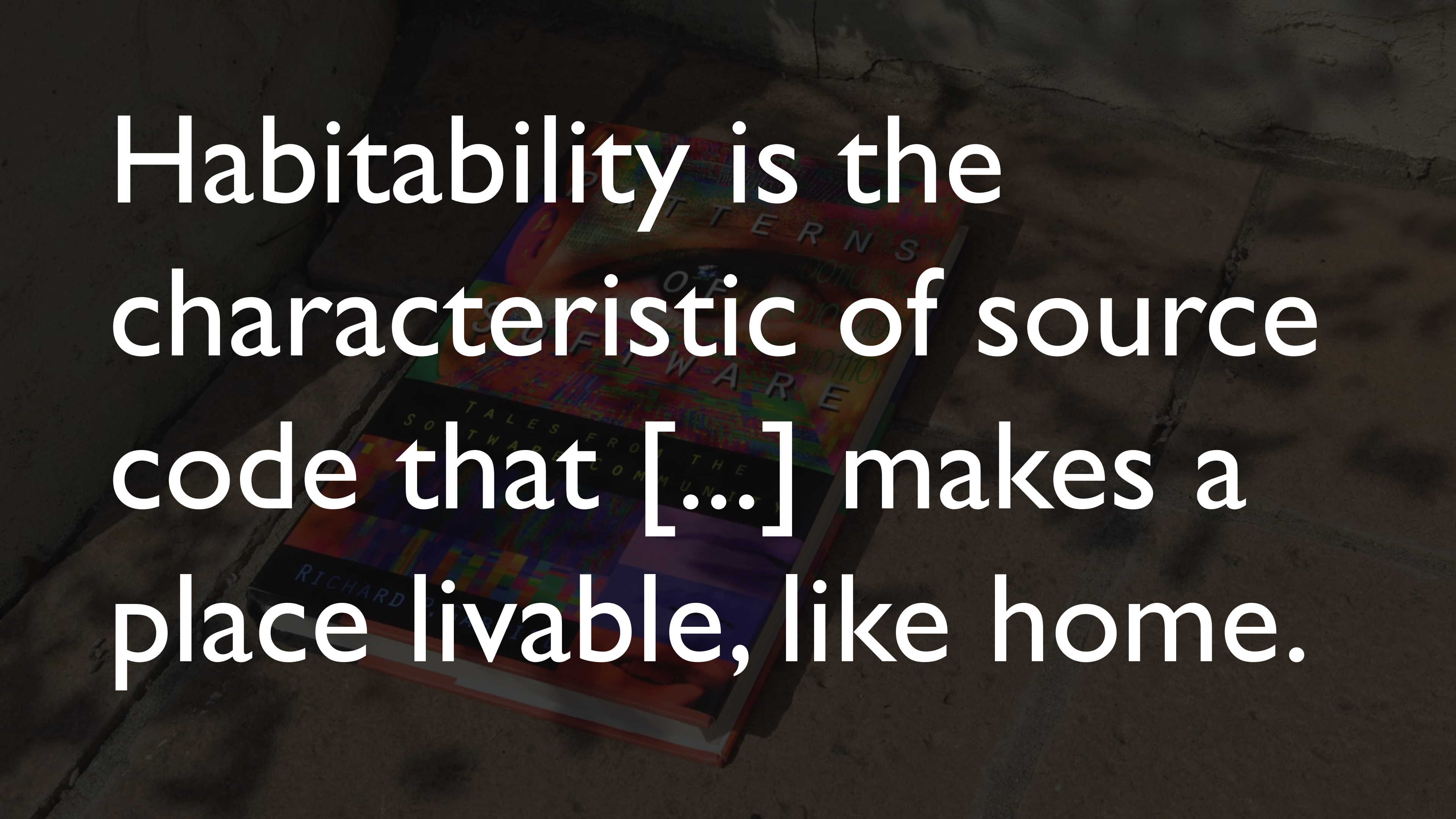
We choose to go to the Moon in this decade and do the other things, not because they are easy, but because they are hard; because that goal will serve to organize and measure the best of our energies and skills, because that challenge is one that we are willing to accept, one we are unwilling to postpone, and one we intend to win, and the others, too.

John F Kennedy

life

```
GAAAAAAGAGGCTGGAAACAAAGGGGTAGAAATGGGGTAGCAGGGGAGATTGCCTGATCAACTGCCAAATG
GTACACAGTTCTGGAAAAGCACAAAAAATGTGCACACACGGGTTCTTCCCACTTTAACCCCTGAGGAATCT
GAGGCCTGCTCCTGAAACAGACTGGGCAGTGGCTAGTGACTCTAGGTATAGGAGTATCCAGCCCtgctcac
ccaggctagagcttAGGGGGACAAGAGGAAAGAGGTGCCTGTGGGGGTGGAGGACAGGAAGGAAAAACACT
CCTGGAATTGCAAAGTGAGGGCAGAGTCTATTTATATTGGGTTTAATTAACTCCTCTCCCTGGTGCCACTA
AAGCAGCAATCACACTGCAGACAGCACTGATTTGATTGGCAAGAGATGCACCAGGCAGAATATTAAGGGAC
CAGGCCCCTATAAATAGGCCTAATCACAGCCCCTCACTGGAAAATGGTAAGGAAGACATTAATCAGGCCTG
GCACTGTGCCCTAGACCTGCTCCCCTAGGCACTACAGTGGGGCCCTTGGTTGCAACACAAGTAGGTAGGGA
TGGATGAGTGTGGCATGAAGGGCCTAGGAGATTTCACTTGGGTTTAAAATgctgtgaccttgagtaagttg
cCGTCTCTGAATCTGATCCTTTCGATTTCCCATTCTCCAAACTGAGAACTAGCACTGCTGAGACGTGGTTA
TTTCcaataataatttgtatattttacataacGCACCACACCAACATCTTCACCCAGTTGGAGCCTACTCC
TTTGCTCCCGCTGCTGGCTTCCCCAGCCCTCCCTTCTGCCCTCCTCAGGCCAGCACTTTTCAGTGAGTTCC
TCCTTTGCATACAGGCTTTCCAGATCTGTACTTGCCTTGAATACTCATCAGAGCCCAGGAGTTACTCCTCA
CCTCCCACTTATTTTTCCTCCCATCAAATAACTAAAGCATGGCCAGCTGATGCCCAGCCAACTGAGAAACC
CAACCCTCTGAGACCAGCACACCCCTTTCAAGCAtgttcctccctccccttctttgTATTTATACTGATGC
AAGTTTGCTGGCTGTCCTAACttatttctgtgcctcagttctccCATATGTAAGATCACAAAGGGGGTAAA
GATGCAAGATATTTCCTGTGCACATCTTCAGATGAATTTcttgttagtgtgtgtgtgtttgctcaCACATA
TGCGTGAAAGAAGAGTACATACACAGATCTCCTCAAAAAGGAGGCAGCAAGCCCGTTCAAGAATGGGACTG
AATACACCTGATGAGTGGTTTACTTTCTGTCTGCAAACATCTACTGATCATCTGTTAGGTGCAGGCCATGA
TCACAACAAAGACGAATAAGACACTACACTAGCCAGGGAGAGTCTCAAAAACAACTAAACTCAAATTAAAT
TCATTCTACTCCAGTCATGGGTACAAAGCTAAGGAGTGACAAATCCCTCTTGGAGTTAGGGGAGTCAGGAA
AAAGCTCTTAGCAGAATGTGTGCCtctcggccgggcgcagcggctcacgcctgtaatcccagcactttggg
```


PATTERNS OF SOFTWARE
TALES FROM THE SOFTWARE COMMUNITY
RICHARD P. GABRIEL

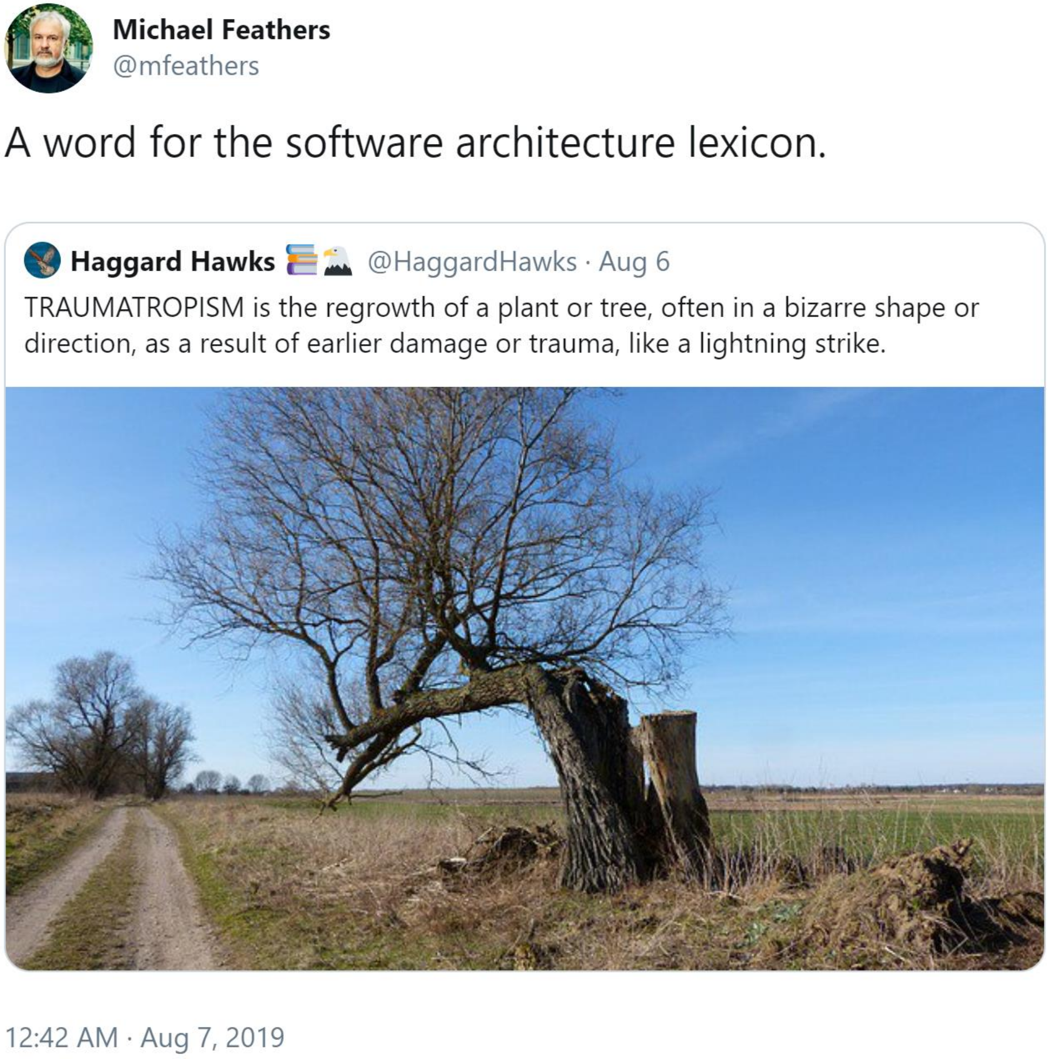


Habitability is the characteristic of source code that [...] makes a place livable, like home.

101 Things I Learned
in Architecture School
Matthew Frederick

Design an architectural space to accommodate a *specific* program, experience, or intent.

POLICE



Michael Feathers
@mfeathers

A word for the software architecture lexicon.

Haggard Hawks 📚🦅 @HaggardHawks · Aug 6

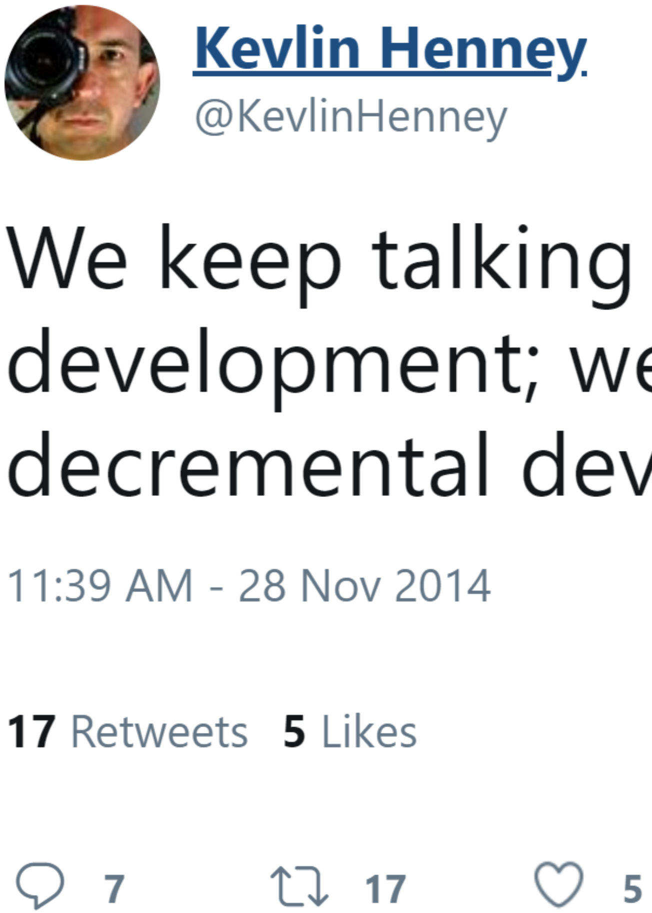
TRAUMATROPISM is the regrowth of a plant or tree, often in a bizarre shape or direction, as a result of earlier damage or trauma, like a lightning strike.

12:42 AM · Aug 7, 2019

https://twitter.com/mfeathers/status/1158870969626218496

Growth for the sake of growth is the ideology of the cancer cell.

Edward Abbey



Kevlin Henney
@KevlinHenney

We keep talking about incremental development; we don't talk enough about decremental development.

11:39 AM - 28 Nov 2014

17 Retweets **5** Likes

https://twitter.com/KevlinHenney/status/538296211166740480

To me programming is more than an important practical art. It is also a gigantic undertaking in the foundations of knowledge.

Grace Hopper

knowledge

know

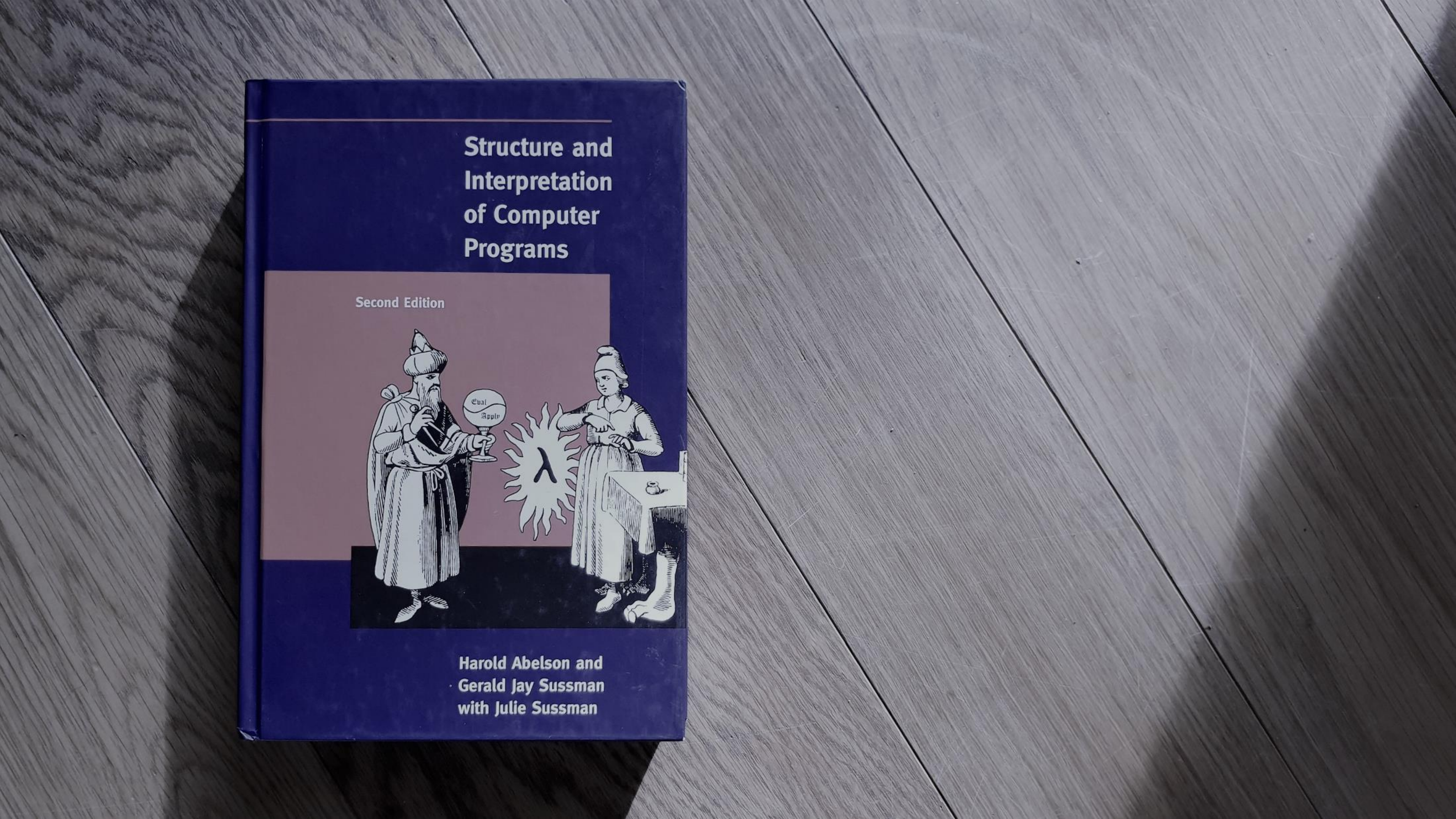
knowability

AN UNSOLVABLE PROBLEM OF ELEMENTARY NUMBER THEORY.[1]

By Alonzo Church.

1. Introduction. There is a class of problems of elementary number theory which can be stated in the form that it is required to find an effectively calculable function f of n positive integers, such that $f(x_1, x_2, \cdot \cdot \cdot, x_n) = 2$[2] is a necessary and sufficient condition for the truth of a certain proposition of elementary number theory involving $x_1, x_2, \cdot \cdot \cdot, x_n$ as free variables.

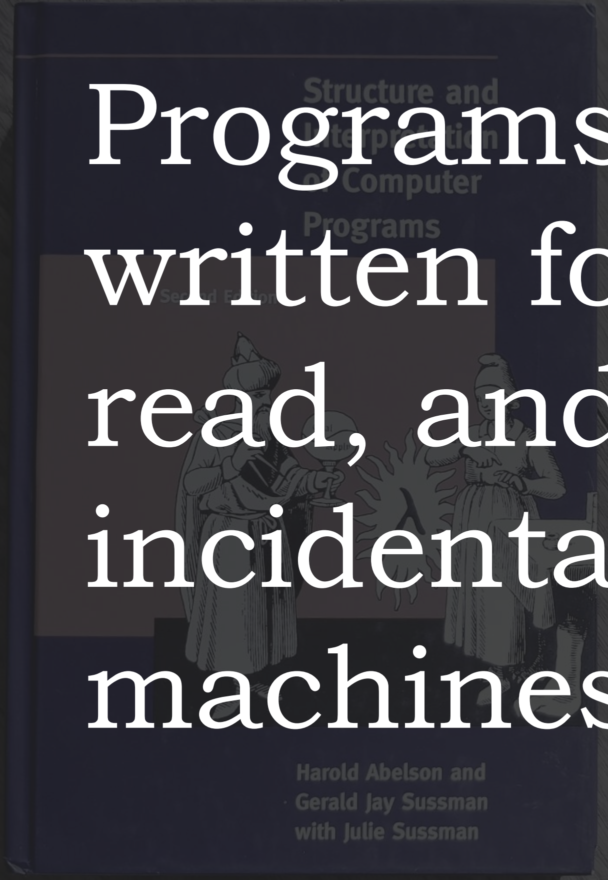
An example of such a problem is the problem to find a means of determining of any given positive integer n whether or not there exist positive integers x, y, z, such that $x^n + y^n = z^n$. For this may be interpreted, required to find an effectively calculable function f, such that $f(n)$ is equal to 2 if and only if there exist positive integers x, y, z, such that $x^n + y^n = z^n$. Clearly



Structure and Interpretation of Computer Programs

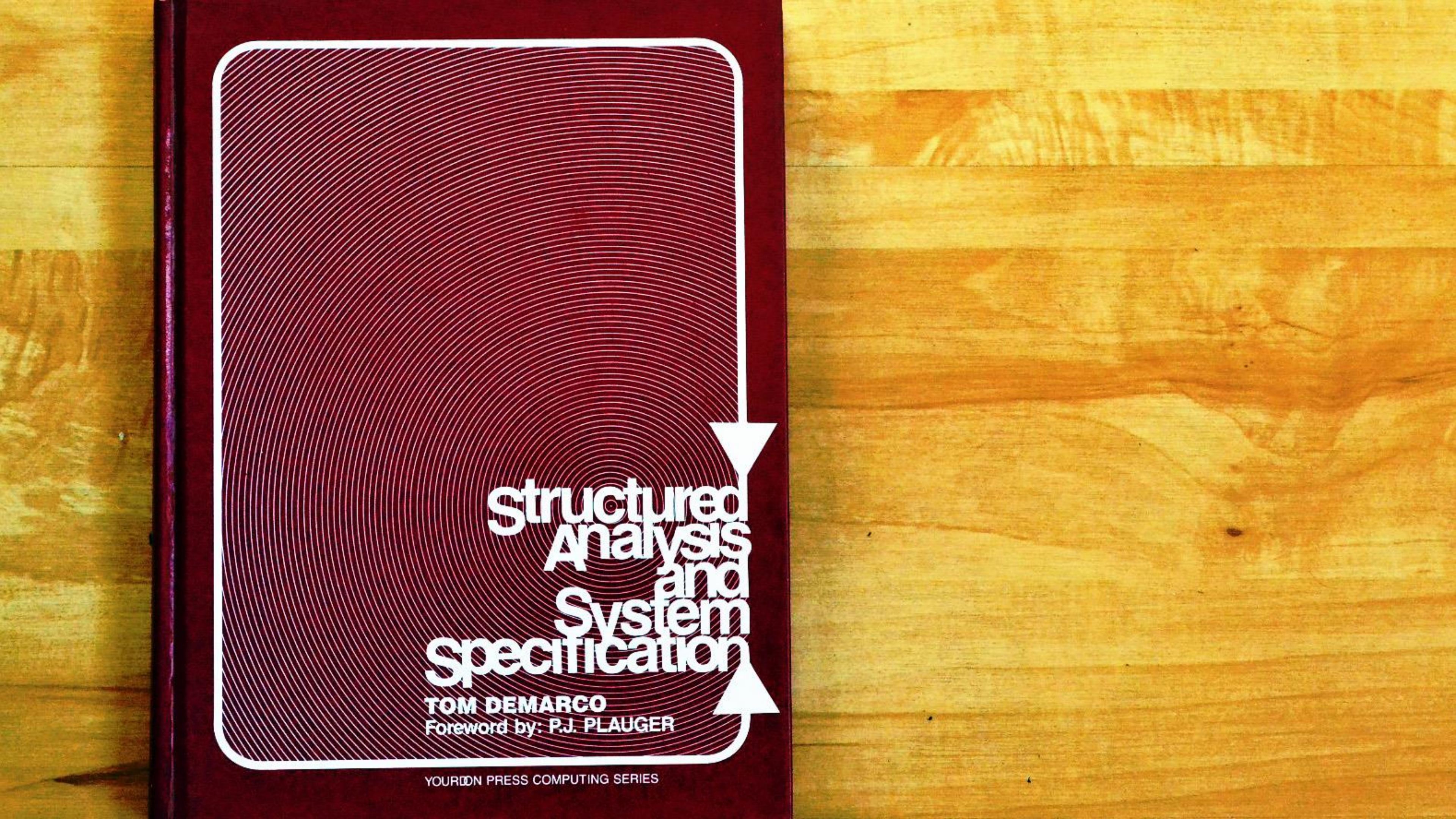
Second Edition

Harold Abelson and
Gerald Jay Sussman
with Julie Sussman



Structure and Interpretation of Computer Programs
Second Edition
Harold Abelson and Gerald Jay Sussman with Julie Sussman

Programs must be written for people to read, and only incidentally for machines to execute.

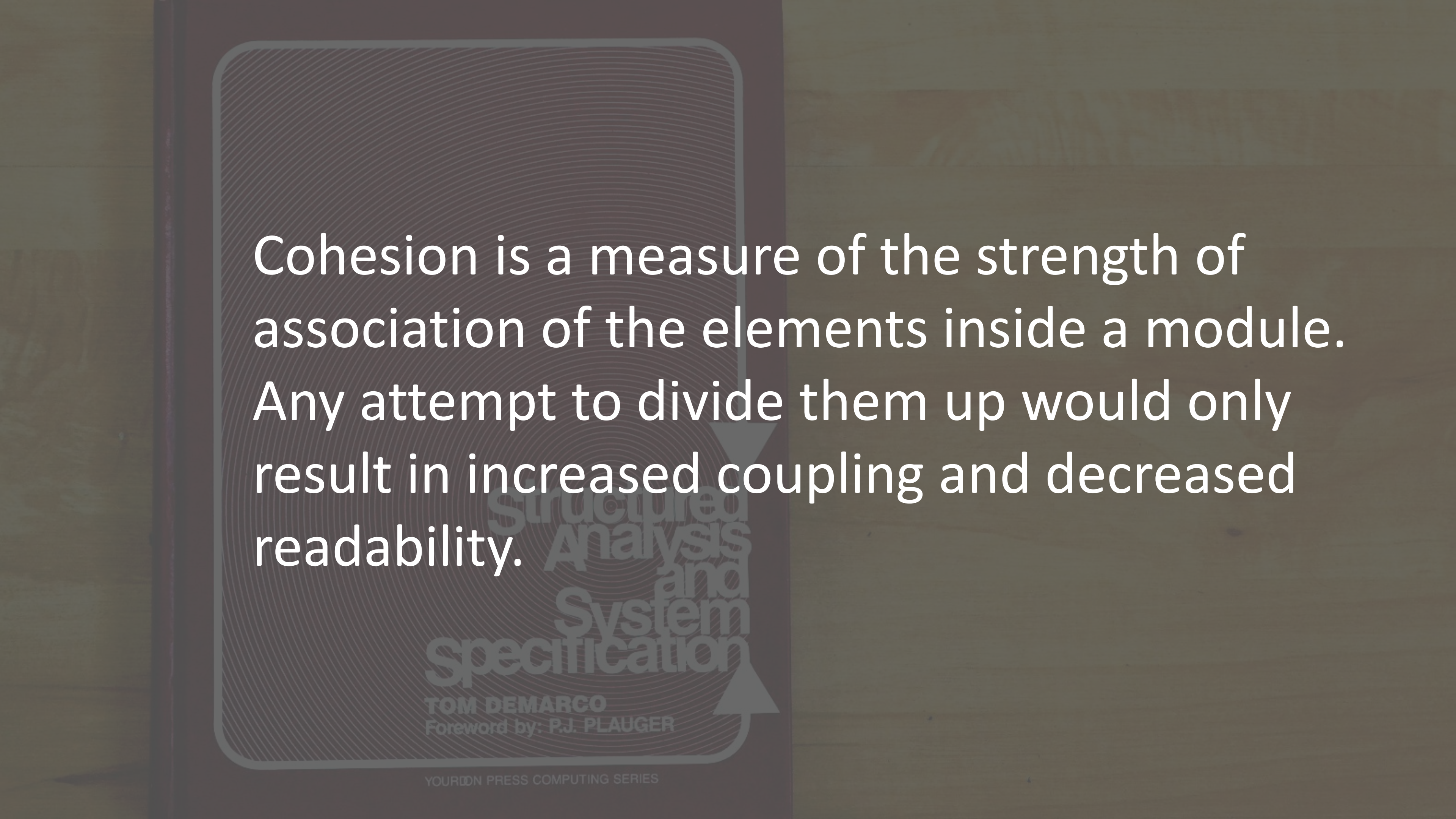


Structured
Analysis
and
System
Specification

TOM DEMARCO
Foreword by: P.J. PLAUGER

YOURDON PRESS COMPUTING SERIES

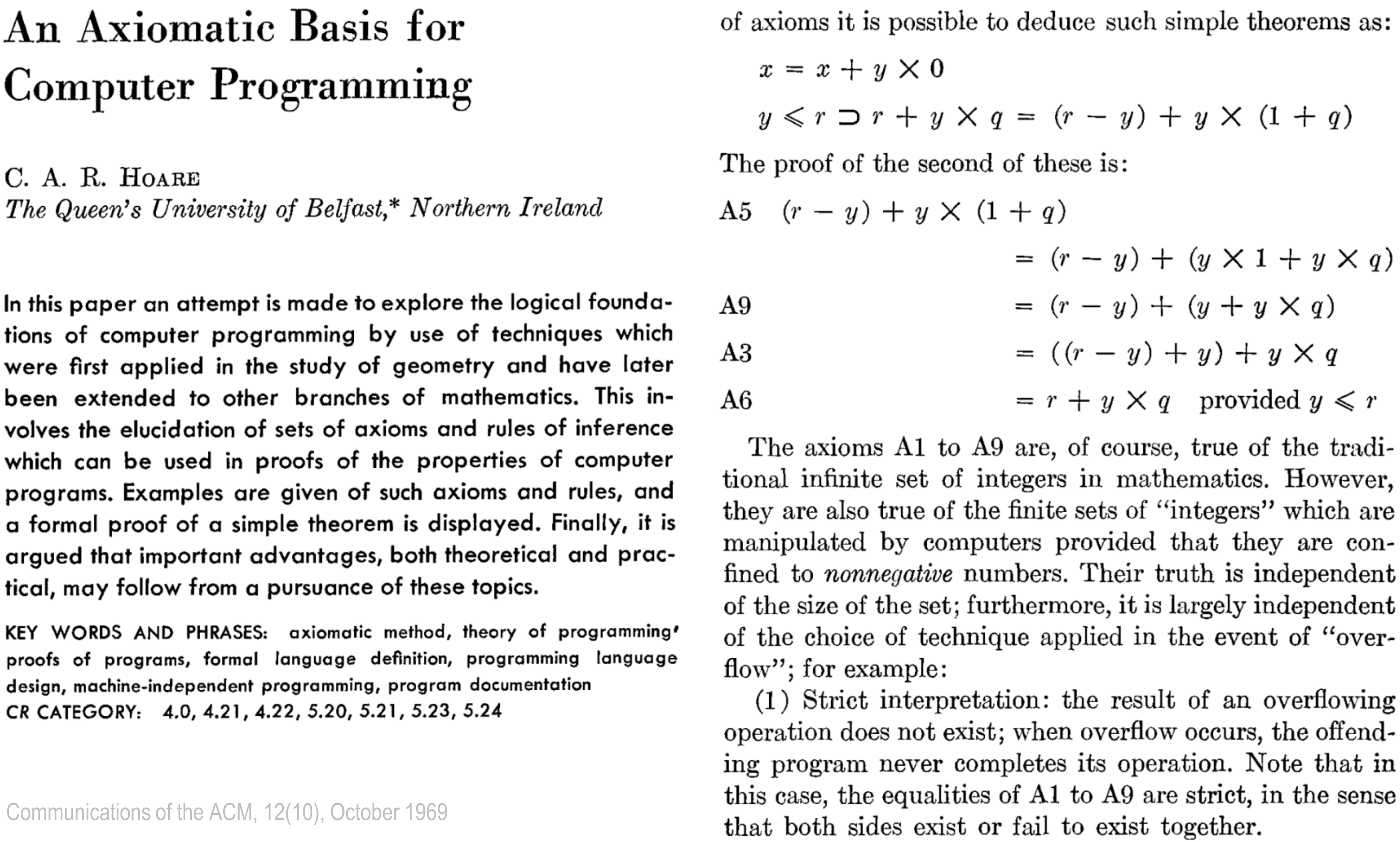
Cohesion is a measure of the strength of association of the elements inside a module. Any attempt to divide them up would only result in increased coupling and decreased readability.


Structured Analysis and System Specification
TOM DEMARCO
Foreword by: P.J. PLAUGER
YOURDON PRESS COMPUTING SERIES

readable

reasonable

reason



An Axiomatic Basis for Computer Programming

C. A. R. Hoare
The Queen's University of Belfast, Northern Ireland*

In this paper an attempt is made to explore the logical foundations of computer programming by use of techniques which were first applied in the study of geometry and have later been extended to other branches of mathematics. This involves the elucidation of sets of axioms and rules of inference which can be used in proofs of the properties of computer programs. Examples are given of such axioms and rules, and a formal proof of a simple theorem is displayed. Finally, it is argued that important advantages, both theoretical and practical, may follow from a pursuance of these topics.

KEY WORDS AND PHRASES: axiomatic method, theory of programming' proofs of programs, formal language definition, programming language design, machine-independent programming, program documentation
CR CATEGORY: 4.0, 4.21, 4.22, 5.20, 5.21, 5.23, 5.24

of axioms it is possible to deduce such simple theorems as:

$$x = x + y \times 0$$

$$y \leqslant r \supset r + y \times q = (r - y) + y \times (1 + q)$$

The proof of the second of these is:

$$\begin{array}{ll}
\text{A5} & (r - y) + y \times (1 + q) \\
 & \quad = (r - y) + (y \times 1 + y \times q) \\
\text{A9} & \quad = (r - y) + (y + y \times q) \\
\text{A3} & \quad = ((r - y) + y) + y \times q \\
\text{A6} & \quad = r + y \times q \quad \text{provided } y \leqslant r
\end{array}$$

The axioms A1 to A9 are, of course, true of the traditional infinite set of integers in mathematics. However, they are also true of the finite sets of "integers" which are manipulated by computers provided that they are confined to *nonnegative* numbers. Their truth is independent of the size of the set; furthermore, it is largely independent of the choice of technique applied in the event of "overflow"; for example:

(1) Strict interpretation: the result of an overflowing operation does not exist; when overflow occurs, the offending program never completes its operation. Note that in this case, the equalities of A1 to A9 are strict, in the sense that both sides exist or fail to exist together.

Communications of the ACM, 12(10), October 1969

programming pearls

By Jon Bentley

WRITING CORRECT PROGRAMS

In the late 1960s people were talking about the promise of programs that verify the correctness of other programs. Unfortunately, it is now the middle of the 1980s, and, with precious few exceptions, there is still little more than talk about automated verification systems. Despite unrealized expectations, however, the research on program verification has given us something far more valuable than a black box that gobbles programs and flashes "good" or "bad"—we now have a fundamental understanding of computer programming.

The purpose of this column is to show how that fundamental understanding can help programmers write correct programs. But before we get to the subject itself, we must keep it in perspective. Coding skill is just one small part of writing correct programs. The majority of the task is the subject of the three previous columns: problem definition, algorithm design, and data structure selection. If you perform those tasks well, then writing correct code is usually easy.

The Challenge of Binary Search

I've given this problem as an in-class assignment in courses at Bell Labs and IBM. The professional programmers had one hour (sometimes more) to convert the above description into a program in the language of their choice; a high-level pseudocode was fine. At the end of the specified time, almost all the programmers reported that they had correct code for the task. We would then take 30 minutes to examine their code, which the programmers did with test cases. In many different classes and with over a hundred programmers, the results varied little: 90 percent of the programmers found bugs in their code (and I wasn't always convinced of the correctness of the code in which no bugs were found).

I found this amazing: only about 10 percent of professional programmers were able to get this small program right. But they aren't the only ones to find this task difficult. In the history in Section 6.2.1 of his *Sorting and Searching*, Knuth points out that while the first binary search was published in 1946, the first published binary search without bugs did not appear until 1962.

Communications of the ACM, 26(12), December 1983

```
public static int binarySearch(int[] a, int key) {
    int low = 0;
    int high = a.length - 1;

    while (low <= high) {
        int mid = (low + high) / 2;
        int midVal = a[mid];

        if (midVal < key)
            low = mid + 1
        else if (midVal > key)
            high = mid - 1;
        else
            return mid; // key found
    }
    return -(low + 1);  // key not found.
}
```

https://ai.googleblog.com/2006/06/extra-extra-read-all-about-it-nearly.html

```
public static int binarySearch(int[] a, int key) {
    int low = 0;
    int high = a.length - 1;

    while (low <= high) {
        int mid = (low + high) / 2;
        int midVal = a[mid];

        if (midVal < key)
            low = mid + 1
        else if (midVal > key)
            high = mid - 1;
        else
            return mid; // key found
    }
    return -(low + 1);  // key not found.
}
```

https://ai.googleblog.com/2006/06/extra-extra-read-all-about-it-nearly.html

```
public static int binarySearch(int[] a, int key) {
    int low = 0;
    int high = a.length - 1;

    while (low <= high) {
        int mid = low + ((high - low) / 2);
        int midVal = a[mid];

        if (midVal < key)
            low = mid + 1
        else if (midVal > key)
            high = mid - 1;
        else
            return mid; // key found
    }
    return -(low + 1);  // key not found.
}
```

https://ai.googleblog.com/2006/06/extra-extra-read-all-about-it-nearly.html

S-Programs

P-Programs

E-Programs

Meir M Lehman

“Programs, Life Cycles, and Laws of Software Evolution”

S-Programs

Programs whose function is formally defined by and derivable from a specification.

Meir M Lehman

“Programs, Life Cycles, and Laws of Software Evolution”

P-Programs

Despite the fact that the problem to be solved can be precisely defined, the acceptability of a solution is determined by the environment in which it is embedded.

Meir M Lehman

"Programs, Life Cycles, and Laws of Software Evolution"

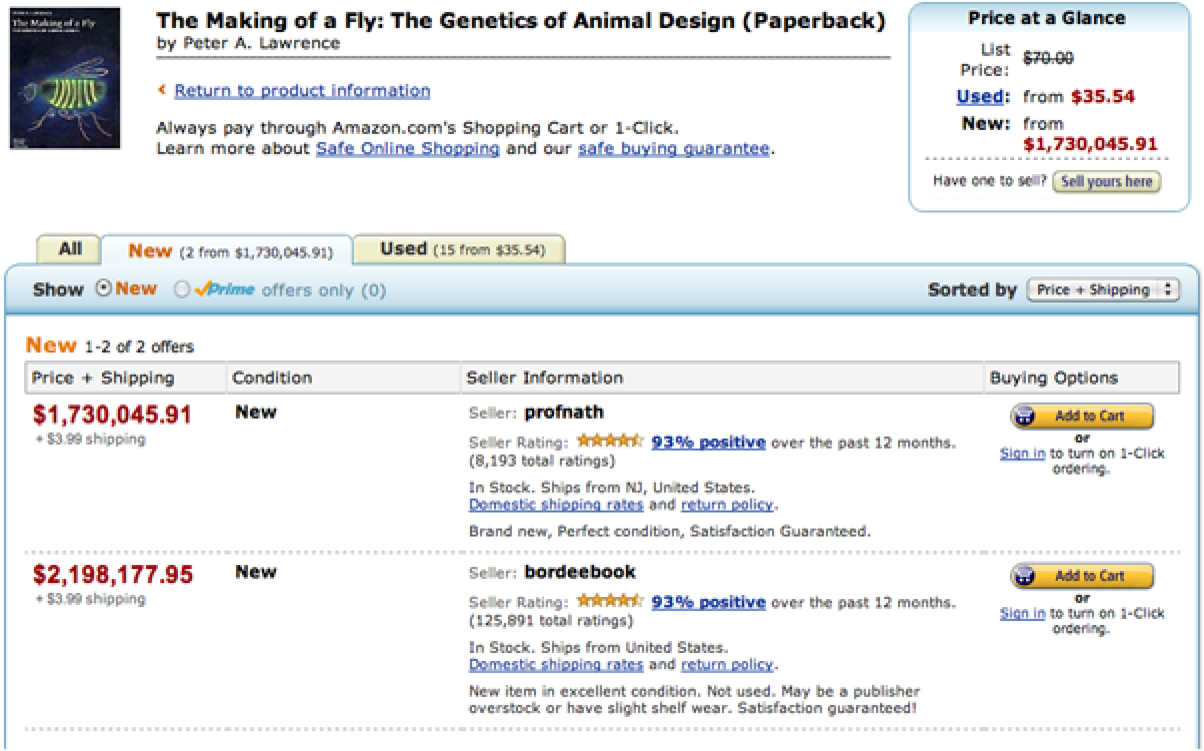
E-Programs

Programs that mechanize a human or societal activity.

The program has become a part of the world it models, it is embedded in it.

Meir M Lehman

"Programs, Life Cycles, and Laws of Software Evolution"



The Making of a Fly: The Genetics of Animal Design (Paperback)
by Peter A. Lawrence

‹ Return to product information

Always pay through Amazon.com's Shopping Cart or 1-Click.
Learn more about Safe Online Shopping and our safe buying guarantee.

Price at a Glance

List Price: ~~$70.00~~
Used: from **$35.54**
New: from **$1,730,045.91**

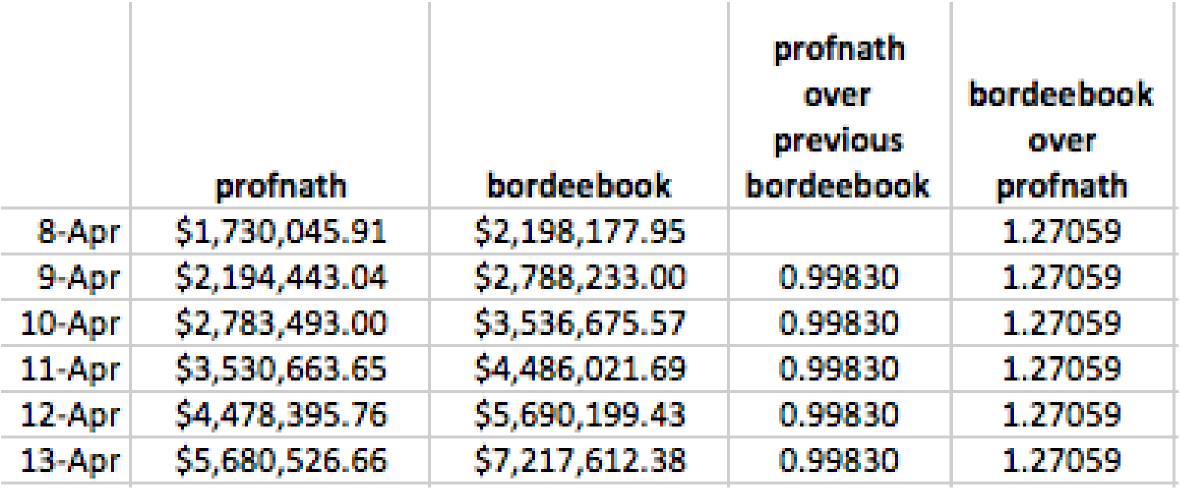
Have one to sell? Sell yours here

Show ● New ○ ✓Prime offers only (0) **Sorted by** Price + Shipping

New 1-2 of 2 offers

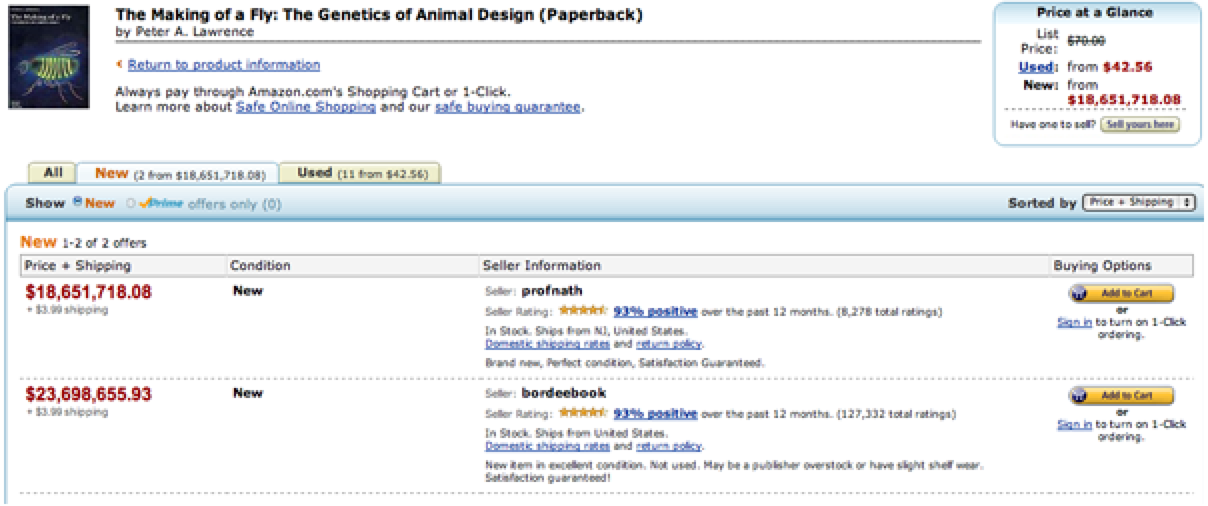
Price + Shipping	Condition	Seller Information	Buying Options
$1,730,045.91 + $3.99 shipping	**New**	Seller: **profnath** Seller Rating: ★★★★☆ 93% positive over the past 12 months. (8,193 total ratings) In Stock. Ships from NJ, United States. Domestic shipping rates and return policy. Brand new, Perfect condition, Satisfaction Guaranteed.	Add to Cart or Sign in to turn on 1-Click ordering.
$2,198,177.95 + $3.99 shipping	**New**	Seller: **bordeebook** Seller Rating: ★★★★☆ 93% positive over the past 12 months. (125,891 total ratings) In Stock. Ships from United States. Domestic shipping rates and return policy. New item in excellent condition. Not used. May be a publisher overstock or have slight shelf wear. Satisfaction guaranteed!	Add to Cart or Sign in to turn on 1-Click ordering.

http://www.michaeleisen.org/blog/?p=358



	profnath	bordeebook	profnath over previous bordeebook	bordeebook over profnath
8-Apr	$1,730,045.91	$2,198,177.95		1.27059
9-Apr	$2,194,443.04	$2,788,233.00	0.99830	1.27059
10-Apr	$2,783,493.00	$3,536,675.57	0.99830	1.27059
11-Apr	$3,530,663.65	$4,486,021.69	0.99830	1.27059
12-Apr	$4,478,395.76	$5,690,199.43	0.99830	1.27059
13-Apr	$5,680,526.66	$7,217,612.38	0.99830	1.27059

http://www.michaeleisen.org/blog/?p=358



The Making of a Fly: The Genetics of Animal Design (Paperback)
by Peter A. Lawrence

Return to product information

Always pay through Amazon.com's Shopping Cart or 1-Click.
Learn more about Safe Online Shopping and our safe buying guarantee.

Price at a Glance

List Price:	~~$70.00~~
Used:	from $42.56
New:	from $18,651,718.08

Have one to sell? Sell yours here

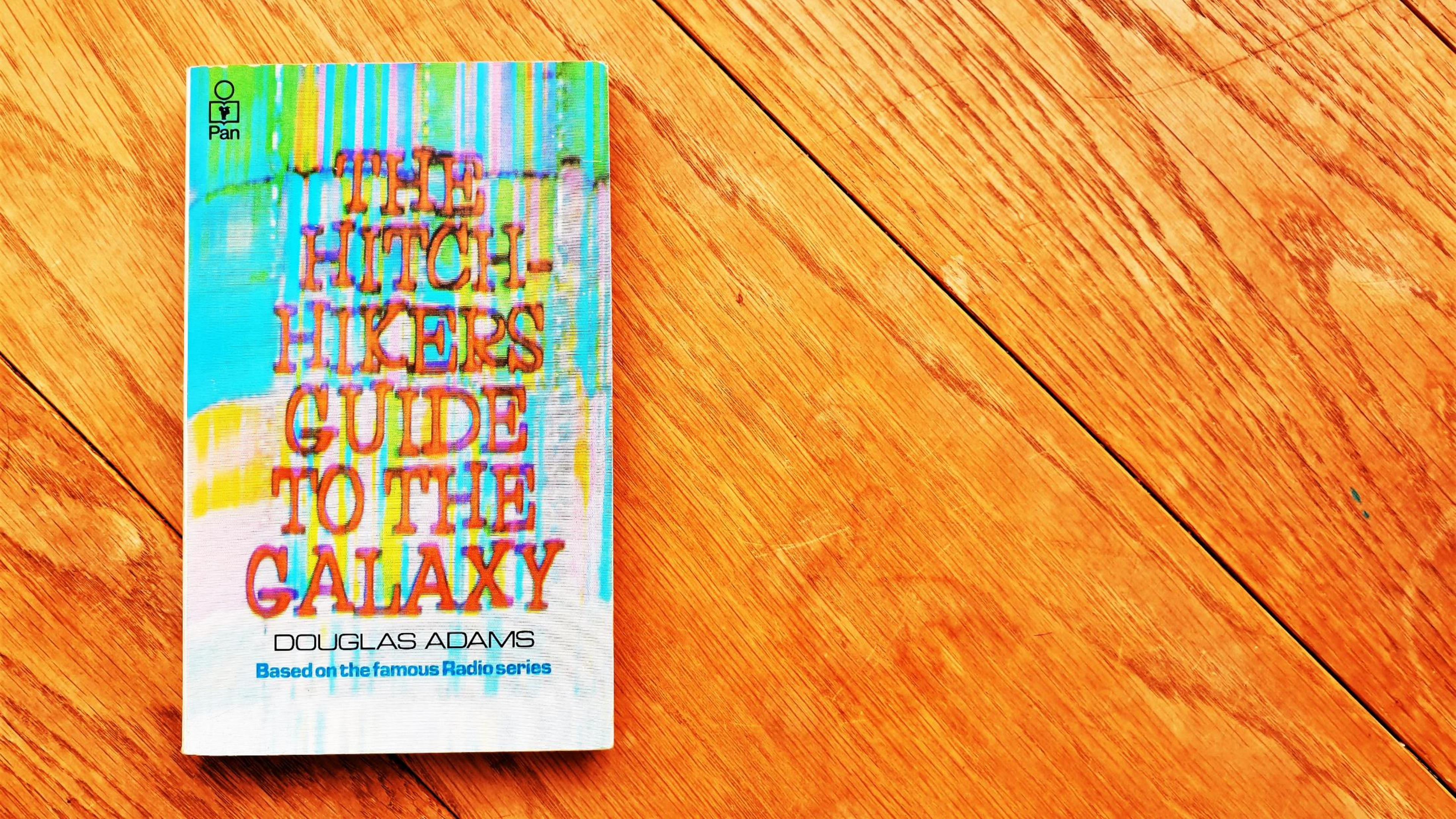
All | New (2 from $18,651,718.08) | Used (11 from $42.56)

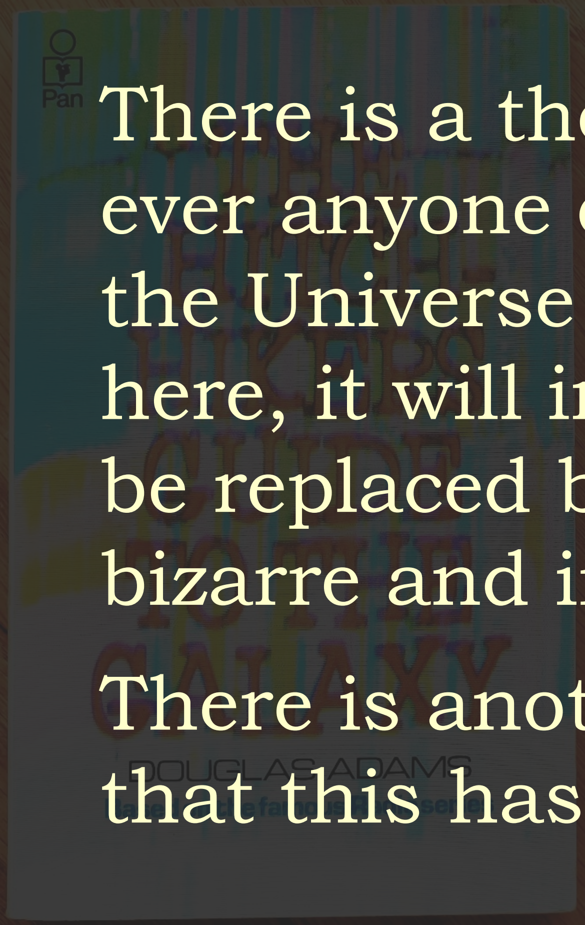
Show ● New ○ Prime offers only (0)   Sorted by Price + Shipping

New 1-2 of 2 offers

Price + Shipping	Condition	Seller Information	Buying Options
$18,651,718.08 + $3.99 shipping	New	Seller: **profnath** Seller Rating: ★★★★½ 93% positive over the past 12 months. (8,278 total ratings) In Stock. Ships from NJ, United States. Domestic shipping rates and return policy. Brand new, Perfect condition, Satisfaction Guaranteed.	Add to Cart or Sign in to turn on 1-Click ordering.
$23,698,655.93 + $3.99 shipping	New	Seller: **bordeebook** Seller Rating: ★★★★½ 93% positive over the past 12 months. (127,332 total ratings) In Stock. Ships from United States. Domestic shipping rates and return policy. New item in excellent condition. Not used. May be a publisher overstock or have slight shelf wear. Satisfaction guaranteed!	Add to Cart or Sign in to turn on 1-Click ordering.

http://www.michaeleisen.org/blog/?p=358


Pan
THE HITCH-HIKERS GUIDE TO THE GALAXY
DOUGLAS ADAMS
Based on the famous Radio series

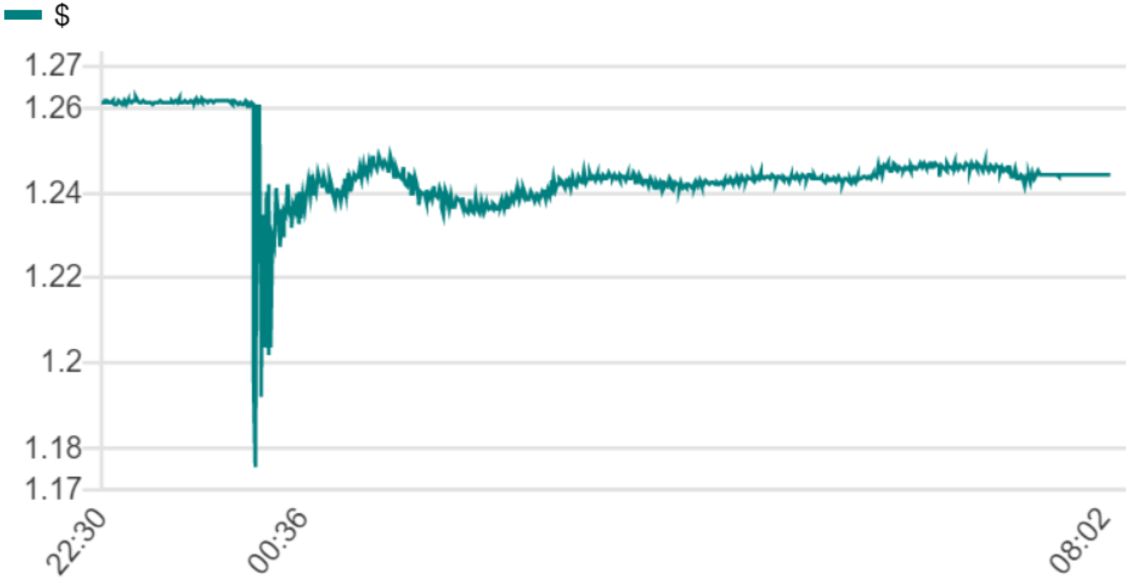


There is a theory which states that if ever anyone discovers exactly what the Universe is for and why it is here, it will instantly disappear and be replaced by something even more bizarre and inexplicable.

There is another theory which states that this has already happened.

Sterling flash crash

£/$, 6-7 October



Source: Bloomberg

BBC

The pound has dived on Asian markets with automated trading being blamed for the volatility.

http://www.bbc.co.uk/news/business-37582150

In code you make choices about how to represent and organise the world of concepts in your software

Kevlin Henney

http://wrywriter.com/qa-with-kevlin-henney/

In code you make choices about how to represent and organise the world of concepts in your software, you create an elaborate and executable fiction based on a number of conceits

Kevlin Henney

http://wrywriter.com/qa-with-kevlin-henney/

In code you make choices about how to represent and organise the world of concepts in your software, you create an elaborate and executable fiction based on a number of conceits, some of which are dictated by programming language and some of which come from the metaphors you choose to imagine and describe how the software should work.

Kevlin Henney

http://wrywriter.com/qa-with-kevlin-henney/

executable fiction

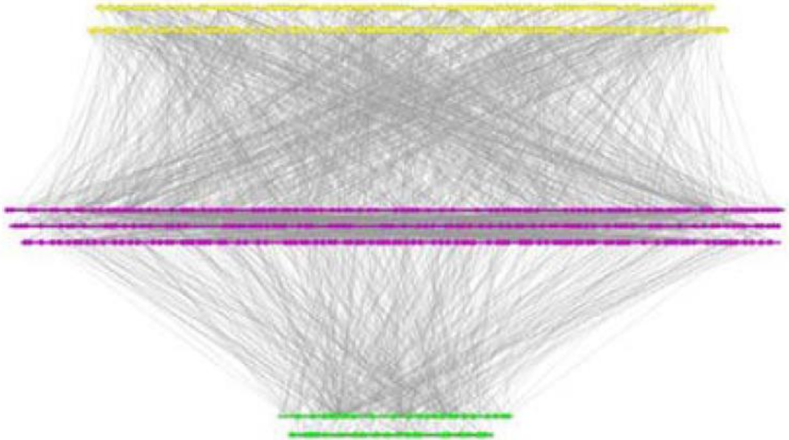
Kevlin Henney

http://wrywriter.com/qa-with-kevlin-henney/

The difference between reality and fiction? Fiction has to make sense.

Tom Clancy

Linux call graph

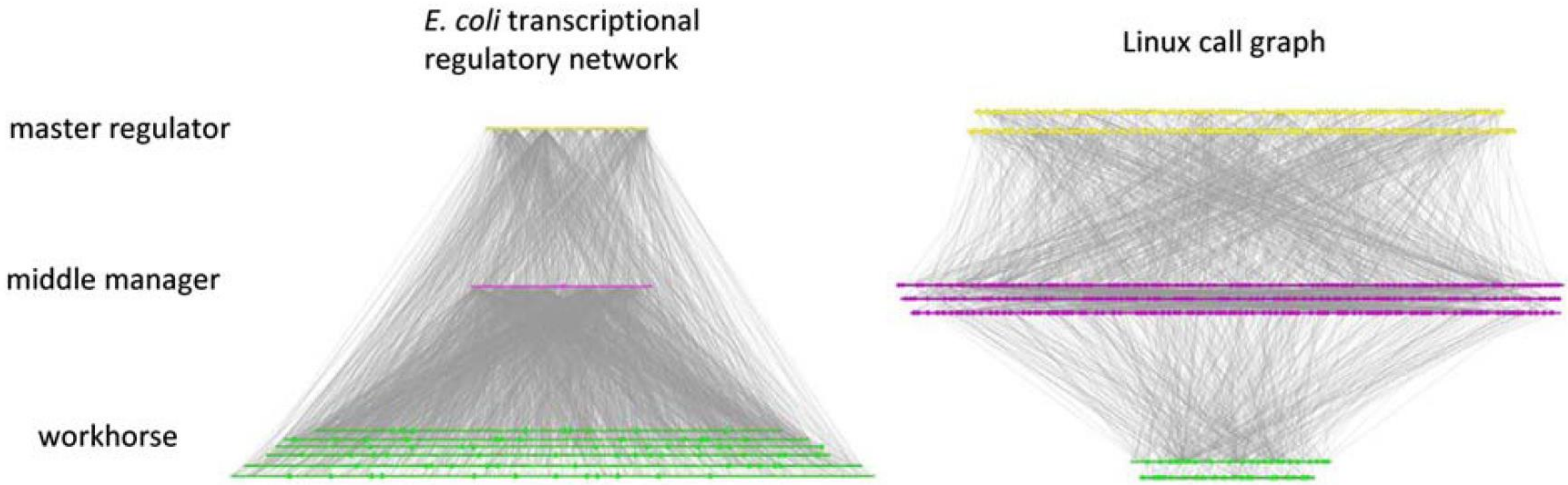


master regulator

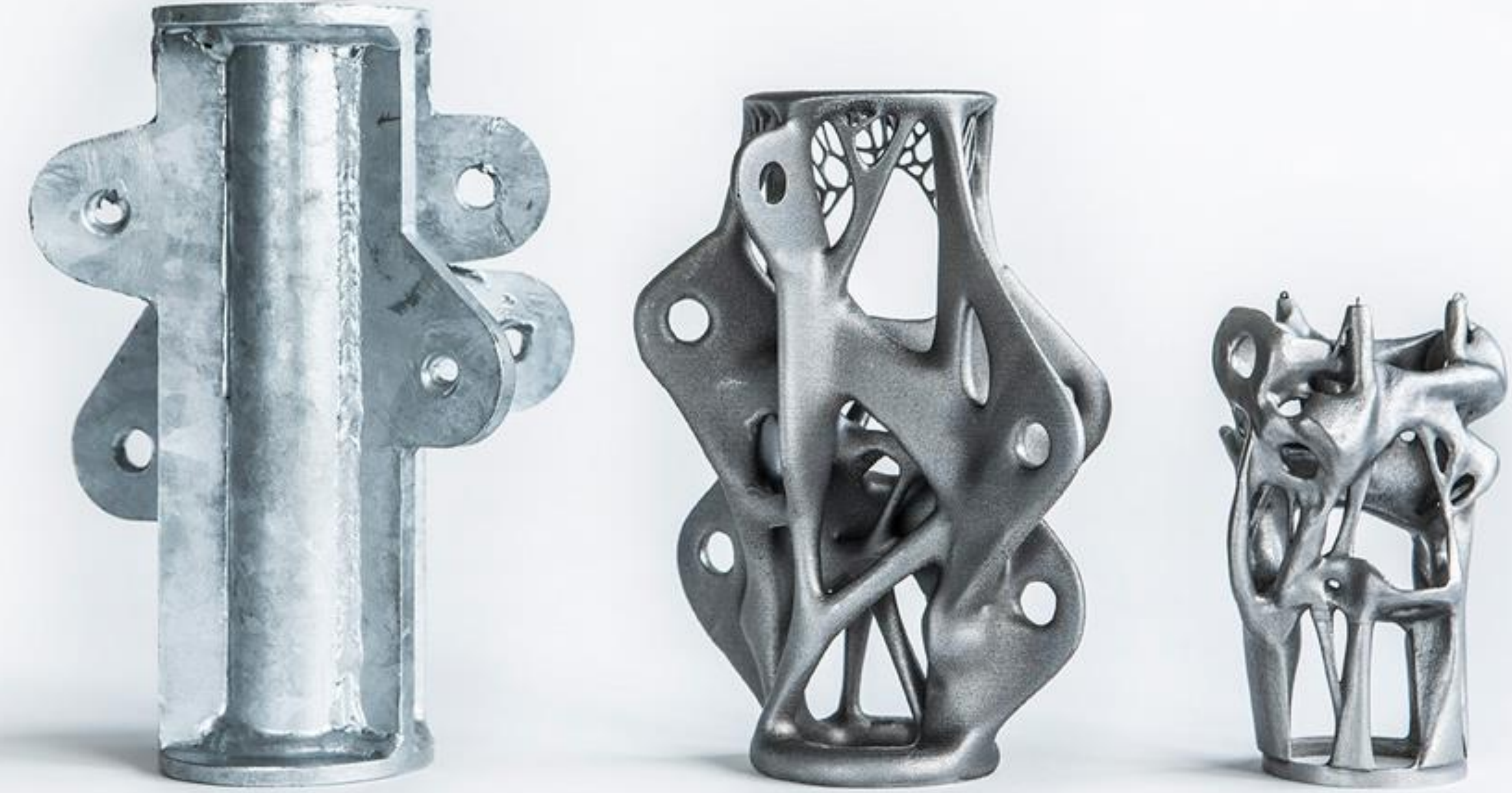
middle manager

workhorse

http://www.pnas.org/content/107/20/9186



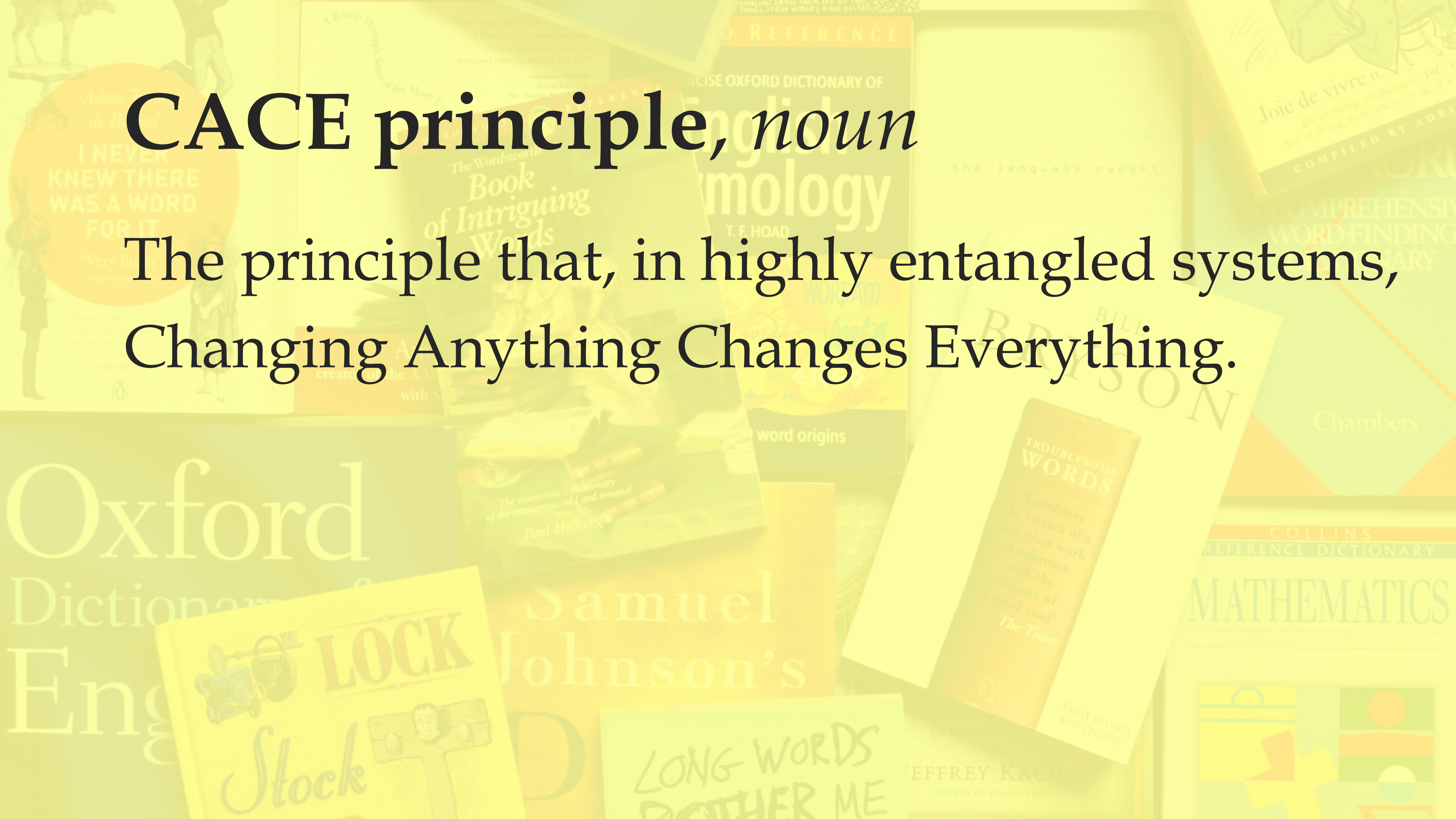
http://www.pnas.org/content/107/20/9186



http://www.arup.com/news/2015_05_may/11_may_3d_makeover_for_hyper-efficient_metalwork

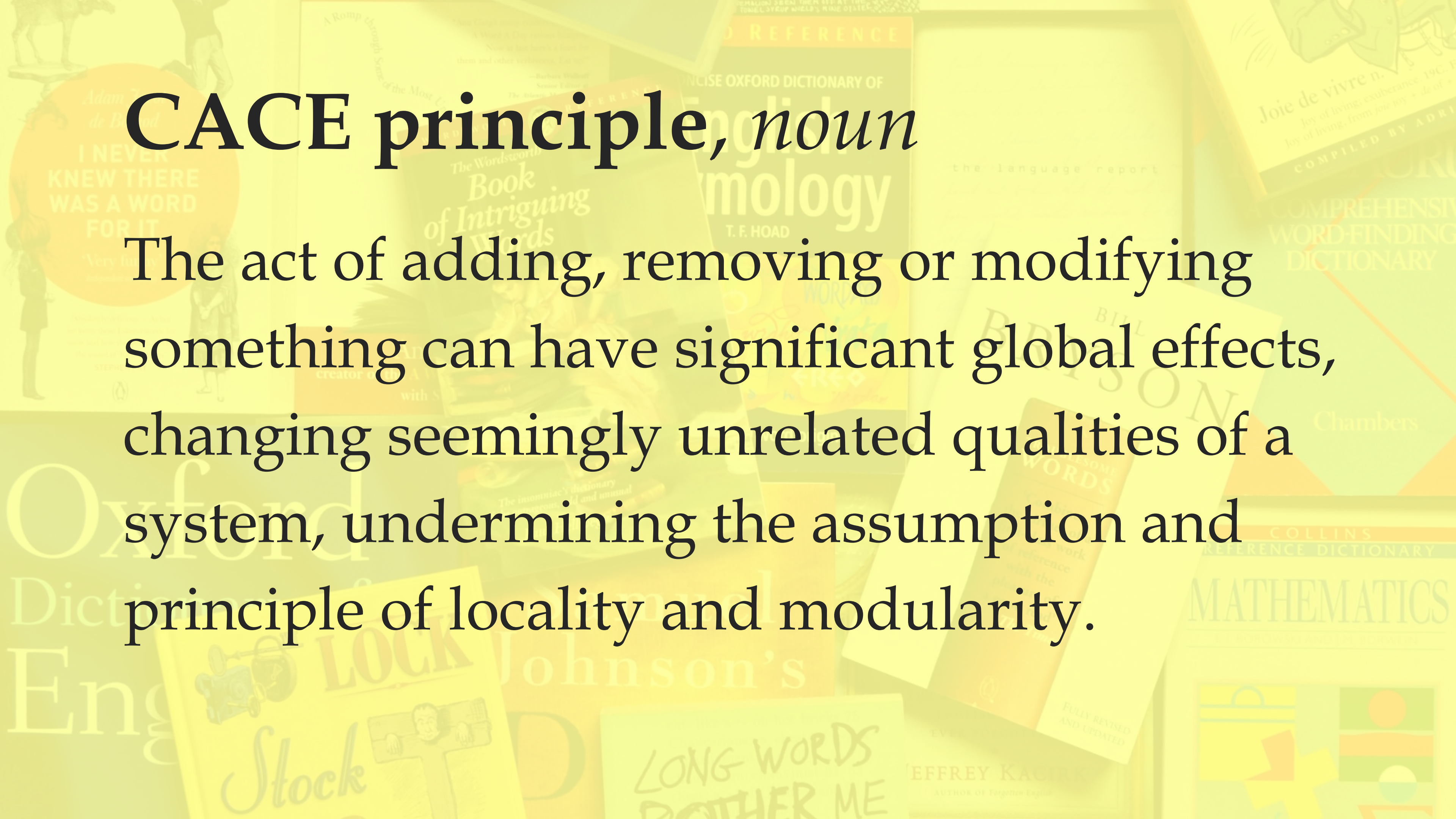
Adam Jacot de Boinod
I NEVER KNEW THERE WAS A WORD FOR IT
'Very funny'
Independent on Sunday
STEPHEN FRY
A Romp through Some of the Most Unusual
WORDSWORTH REFERENCE
The Wordsworth Book of Intriguing Words
The insomniac's dictionary of the outrageous, odd and unusual
Paul Hellweg
CONCISE OXFORD DICTIONARY OF
English Etymology
T. F. HOAD
word origins
the language report
Joie de vivre n.
COMPILED BY ADRIAN
A COMPREHENSIVE WORD-FINDING DICTIONARY
Chambers
BILL BRYSON
TROUBLESOME WORDS
'Combines the virtues of a first-class work of reference with the pleasure of a good read'
The Times
FULLY REVISED AND UPDATED
Oxford Dictionary of English
LOCK Stock
Samuel Johnson's
LONG WORDS BOTHER ME
JEFFREY KACIRK
AUTHOR OF Forgotten English
COLLINS REFERENCE DICTIONARY
MATHEMATICS
E. J. BOROWSKI AND J. M. BORWEIN

f / WordFriday


I NEVER KNEW THERE WAS A WORD FOR IT
The Wordsworth Book of Intriguing Words
CONCISE OXFORD DICTIONARY OF English Etymology
T. F. HOAD
word origins
the language report
Joie de vivre
COMPREHENSIVE WORD-FINDING DICTIONARY
BILL BRYSON
Chambers
Oxford Dictionary of English
LOCK Stock
Samuel Johnson's
TROUBLESOME WORDS
LONG WORDS BOTHER ME
COLLINS REFERENCE DICTIONARY
MATHEMATICS

CACE principle, *noun*

The principle that, in highly entangled systems, Changing Anything Changes Everything.



CACE principle, *noun*

The act of adding, removing or modifying something can have significant global effects, changing seemingly unrelated qualities of a system, undermining the assumption and principle of locality and modularity.

MIND THE GAP