```
Script started on Wed Jul 12 08:47:06 2006
Publications/2006-Dagstuhl-06281/Cousot-Dagstuhl-06281_demonstration % ./README



***************************************************
***************************************************
***                                             ***
***   Demonstration of the Astree static analyzer   ***
***             http://www.astree.ens.fr/           ***
***                                             ***
***************************************************
***************************************************


********************************************
* Astree is SOUND hence reports ALL potential *
* runtime errors                              *
********************************************



*** example [CC76]:


cat -n dichotomy-error.c
     1  /* dichotomy-error.c */
     2  int main () {
     3      int lwb, upb, m, R[100], X;
     4      lwb = 1; upb = 100;
     5      while (lwb <= upb) {
     6         m = (upb + lwb) / 2;
     7         if (X == R[m]) {
     8            upb = m; lwb = m+1; }
     9         else if (X < R[m]) {
    10            upb = m - 1; }
    11         else {
    12            lwb = m + 1; }
    13      }
    14       __ASTREE_log_vars((m));
    15  }


*** static analysis by Astree:


/* Analyzer launched at 2006/ 7/12 06:47:19
dichotomy-error.c:7.15-19::[call#main@2:loop@5>=4:]: WARN: invalid dereference: dereferencing
dichotomy-error.c:7.15-19::[call#main@2:loop@5>=4:]: WARN: invalid dereference: dereferencing
dichotomy-error.c:9.19-23::[call#main@2:loop@5>=4:]: WARN: invalid dereference: dereferencing
dichotomy-error.c:9.19-23::[call#main@2:loop@5>=4:]: WARN: invalid dereference: dereferencing
%

*** (the two errors are reported two times
      each because of partitioning.)



***************************************************
* Astree is INCOMPLETE hence may report false alarms *
***************************************************



*** example:
```

```
cat -n fausse-alarme.c
     1	/* fausse-alarme.c */
     2	void main()
     3	{
     4	  int x, y;
     5	  if ((-4681 < y) && (y < 4681) && (x < 32767) && (-32767 < x) && ((7*y*y - 1) ==
     6	     y = 1 / x;
     7	  };
     8	}
```

```
*** static analysis by Astree:
```

```
astree --exec-fn main --no-relational fausse-alarme.c |& egrep "(launched)|(WARN)"
/* Analyzer launched at 2006/ 7/12 06:47:23
fausse-alarme.c:6.9-14::[call#main@2:]: WARN: integer division by zero ([-32766, 32766] and
%
```

```
**************************************************
* Astree is based upon classical abstract domains *
* such as intervals...                            *
**************************************************
```

```
*** example:
```

```
cat -n dichotomy.c
     1	/* dichotomy.c */
     2	int main () {
     3	    int lwb, upb, m, R[100], X;
     4	    lwb = 0; upb = 99;
     5	    while (lwb <= upb) {
     6	       m = (upb + lwb) / 2;
     7	       if (X == R[m]) {
     8	          upb = m; lwb = m+1; }
     9	       else if (X < R[m]) {
    10	          upb = m - 1; }
    11	       else {
    12	          lwb = m + 1; }
    13	    }
    14	     __ASTREE_log_vars((m));
    15	}
```

```
*** correction (difference with the erroneous version):
```

```
1c1
< /* dichotomy-error.c */
---
> /* dichotomy.c */
4c4
<     lwb = 1; upb = 100;
---
>     lwb = 0; upb = 99;
```

```
*** static analysis by Astree:
```

```
astree --exec-fn main --no-relational dichotomy.c  |& egrep "(launched)|(m in )"
/* Analyzer launched at 2006/ 7/12 06:47:28
   <interval: <m in [0, 99]> <> > <modulo: >
```

```
%
```

*** (pay attention to the case of many global variables!)

```
******************************************
* Astree uses weakly relational abstract *
* domains such as octagons...            *
******************************************
```

*** example:

```
cat -n octagon.c
     1	/* octagon.c */
     2	void main()
     3	{
     4	  int X, Y, Z;
     5	  X = 10;
     6	  Y = 100;
     7	  while (X >= 0) {
     8	    X--;
     9	    Y--;
    10	  };
    11	   __ASTREE_assert((X <= Y));
    12	}
```

*** static analysis by Astree (1 -- WITHOUT octagons):

```
astree octagon.c --no-octagon --exec-fn main |& egrep "(launched)|(WARN)"
/* Analyzer launched at 2006/ 7/12 06:47:31
octagon.c:9.4-7::[call#main@2:loop@7>=4:]: WARN: signed int arithmetic range [-2147483649,
octagon.c:11.19-25::[call#main@2:]: WARN: assert failure
%
```

*** static analysis by Astree (2 -- WITH octagons):

```
astree octagon.c --exec-fn main |& egrep "(launched)|(WARN)"
/* Analyzer launched at 2006/ 7/12 06:47:33
%
```

*** (does not scale up for too many variables!)

```
*********************************************
* Astree uses weakly relational abstract    *
* domains such as boolean decision trees... *
*********************************************
```

*** example:

```
cat -n boolean.c
     1	/* boolean.c */
     2	typedef enum {F=0,T=1} BOOL;
     3	BOOL B;
     4	void main () {
```

```
     5       unsigned int X, Y;
     6       while (1) {
     7          /* ... */
     8          B = (X == 0);
     9          /* ... */
    10          if (!B) {
    11             Y = 1 / X;
    12          }
    13          /* ... */
    14      }
    15   }
```

```
*** static analysis by Astree (1 -- WITHOUT
     decision trees):
```

```
astree boolean.c --no-relational --exec-fn main |& egrep "(launched)|(WARN)"
/* Analyzer launched at 2006/ 7/12 06:47:36
boolean.c:11.13-18::[call#main@4:loop@6=1:]: WARN: integer division by zero ([0, 4294967295
boolean.c:11.13-18::[call#main@4:loop@6=2:]: WARN: integer division by zero ([0, 4294967295
boolean.c:11.13-18::[call#main@4:loop@6=3:]: WARN: integer division by zero ([0, 4294967295
boolean.c:11.13-18::[call#main@4:loop@6>=4:]: WARN: integer division by zero ([0, 429496729
%
```

```
*** static analysis by Astree (2 -- WITH
     decision trees):
```

```
astree boolean.c --exec-fn main |& egrep "(launched)|(WARN)"
/* Analyzer launched at 2006/ 7/12 06:47:38
%
```

```
*********************************************
* Astree uses computation trace abstractions *
* (describing  sequences of states) not only *
* invariants (describing sets of states)     *
*********************************************
```

```
*** example:
```

```
cat -n trace-partitioning.c
     1   /* trace-partitioning.c */
     2   void main() {
     3   float t[5] = {-10.0, -10.0, 0.0, 10.0, 10.0};
     4   float c[4] = {0.0, 2.0, 2.0, 0.0};
     5   float d[4] = {-20.0, -20.0, 0.0, 20.0};
     6   float x, r;
     7   int i = 0;
     8   __ASTREE_known_fact(((-100.0 <= x) && (x <= 100.0)));
     9   while ((i < 3) && (x >= t[i+1])) {
    10   i = i + 1;
    11   }
    12   r = (x - t[i]) * c[i] + d[i];
    13   __ASTREE_log_vars((r));
    14   }
```

```
*** static analysis by Astree (1 -- WITH
     partitioning):
```

```
astree --exec-fn main --no-trace --no-relational trace-partitioning.c |& egrep "(launched)
/* Analyzer launched at 2006/ 7/12 06:47:41
  <interval: <> <r in [-20, 20]> > <modulo: >
%


*** static analysis by Astree (2 -- WITHOUT
     partitioning):


astree --exec-fn main --no-partition --no-trace --no-relational trace-partitioning.c |& eg
/* Analyzer launched at 2006/ 7/12 06:47:42
  <interval: <> <r in [-240, 240]> > <modulo: >
%


*************************************
* Astree handles floats, not reals or *
* fixed point arithmetics             *
*************************************



*** example of computation error in floats:
***    (x+a)-(x-a) <> 2a! with float


cat -n float-float.c
     1  /* float-float.c */
     2  int main () {
     3  float x; float a, y, z, r1, r2;
     4  a = 1.0;
     5  x = 1125899973951488.0;
     6  y = (x + a);
     7  z = (x - a);
     8  r1 = y - z;
     9  r2 = 2 * a;
    10  printf("(x + a) - (x - a) = %f\n", r1);
    11  printf("2a                = %f\n", r2);
    12  }


*** compilation:


float-float.c: In function 'main':
float-float.c:10: warning: incompatible implicit declaration of built-in function 'printf'


*** execution:

(x + a) - (x - a) = 0.000000
2a                = 2.000000

*** more precision can be better...
***    (x+a)-(x-a) <> 2a! with double


cat -n double-double.c
     1  /* double-double.c */
     2  int main () {
     3  double x; double a, y, z, r1, r2;
     4  a = 1.0;
     5  x = 1125899973951488.0;
     6  y = (x + a);
     7  z = (x - a);
     8  r1 = y - z;
     9   r2 = 2 * a;
    10   printf("(x + a) - (x - a) = %f\n", r1);
    11   printf("2a                = %f\n", r2);
    12   }


*** compilation:

double-double.c: In function 'main':
double-double.c:10: warning: incompatible implicit declaration of built-in function 'print


*** execution:


(x + a) - (x - a) = 2.000000
2a                = 2.000000


*** computations with different precisions...
*** can be really catastrophic!
***    (x+a)-(x-a) <> 2a! with double+float


cat -n double-float.c
     1   /* double-float.c */
     2   int main () {
     3   double x; float a, y, z, r1, r2;
     4   a = 1.0;
     5   x  = 1125899973951488.0;
     6   y = (x + a);
     7   z = (x - a);
     8   r1 = y - z;
     9   r2 = 2 * a;
    10   printf("(x + a) - (x - a) = %f\n", r1);
    11   printf("2a                = %f\n", r2);
    12   }


*** compilation:

double-float.c: In function 'main':
double-float.c:10: warning: incompatible implicit declaration of built-in function 'printf


*** execution:


(x + a) - (x - a) = 134217728.000000
2a                = 2.000000


*** testing is unlikely to make it!
***    (x+a)-(x-a) <> 2a! with double+float


cat -n double-float2.c
     1   /* double-float2.c */
     2   int main () {
     3   double x; float a, y, z, r1, r2;
     4   a = 1.0;
     5   x  = 1125899973951487.0;
     6   y = (x + a);
     7   z = (x - a);
     8   r1 = y - z;
     9   r2 = 2 * a;
    10   printf("(x + a) - (x - a) = %f\n", r1);
    11   printf("2a                  = %f\n", r2);
    12   }


*** only one digit difference:


1c1
< /* double-float2.c */
---
> /* double-float.c */
5c5
< x = 1125899973951487.0;
---
> x = 1125899973951488.0;


*** compilation:

double-float2.c: In function 'main':
double-float2.c:10: warning: incompatible implicit declaration of built-in function 'print


*** execution:


(x + a) - (x - a) = 0.000000
2a                = 2.000000


************************************************
* Astree takes rounding errors into account... *
************************************************



*** example ((x+a)-(x-a) <> 2a! in double+float):


cat -n double-float-analyze.c
     1   /* double-float-analyze.c */
     2   int main () {
     3   double x; float a, y, z, r1, r2;
     4   a = 1.0;
     5   x = 1125899973951488.0;
     6   y = (x + a);
     7   z = (x - a);
     8   r1 = y - z;
     9   r2 = 2 * a;
    10   __ASTREE_log_vars((r1, r2));
    11   }


*** static analysis by Astree:

astree --exec-fn main double-float-analyze.c |& egrep "(launched)|(r2 in )|(r1 in)"
/* Analyzer launched at 2006/ 7/12 06:47:58
  <interval: <> <r2 in {2}, r1 in [-1.34218e+08, 1.34218e+08]> > <symbolic: >
%


********************************************************
* Astree takes into account the potential accumulation *
* of rounding errors over very long periods of time... *
********************************************************
```

```
***  example:


cat  -n  bary.c
     1   /* bary.c */
     2   typedef  enum  {FALSE  =  0,  TRUE  =  1}  BOOLEAN;
     3   float  INIT,C1,I;
     4   float RANDOM_INPUT;
     5   __ASTREE_volatile_input((RANDOM_INPUT  [-1.,1.]));
     6
     7   void bary () {
     8     static float X,Y,Z;
     9     if (C1>0.)
    10       {Z = Y;Y = X;}
    11     if (INIT>0.)
    12       {
    13         X=I;
    14         Y=I;
    15         Z=I;
    16       }
    17     else
    18        {X  =  0.50000001  *  X  +  0.30000001*Y  +  0.20000001*Z   ;};
    19      __ASTREE_log_vars((X,Y,Z));
    20
    21   }
    22
    23   void main () {
    24    INIT = 1.;
    25     C1 = RANDOM_INPUT;
    26     I   = RANDOM_INPUT;
    27     while (1) {
    28       bary();
    29       INIT = RANDOM_INPUT;
    30       C1 = RANDOM_INPUT;
    31       I   = RANDOM_INPUT;
    32        __ASTREE_wait_for_clock(());
    33    }
    34   }


***  configuration  file  (10  hours  at  1/100th  s):

cat  -n  bary10.config
     1   __ASTREE_max_clock((3600000));


***  static  analysis  by  Astree:


astree  --exec-fn  main  --config-sem  bary10.config  bary.c  |&  tail  -n  75  |  egrep  --after-cont
   <interval:  <>  <Z  in  [-1.71113,  1.71113],  Y  in  [-1.71113,  1.71113],
    X  in  [-1.71113,  1.71113]>  >  <symbolic:  >
%


***  configuration  file  (100  hours  at  1/100th  s):

cat  -n  bary100.config
     1   __ASTREE_max_clock((36000000));


***  static  analysis  by  Astree:


astree  --exec-fn  main  --config-sem  bary100.config  bary.c  |&  tail  -n  75  |  egrep  --after-cor
   <interval:  <>  <Z  in  [-215.193,  215.193],  Y  in  [-215.193,  215.193],
    X in [-215.193, 215.193]> > <symbolic: >
%
```

```
*** configuration file (1000 hours at 1/100th s):

cat -n bary1000.config
__ASTREE_max_clock((360000000));
```

```
*** static analysis by Astree:


astree --exec-fn main --config-sem bary1000.config bary.c |& tail -n 50 | egrep --after-co
   <interval: <> <Z in [-2.1295e+23, 2.1295e+23],
    Y in [-2.1295e+23, 2.1295e+23], X in [-2.1295e+23, 2.1295e+23]> >
%
```

```
*** (note that the analysis time is independent
      of the execution time.)
```

```
****************************************************
* Astree knows about basic numerical computations... *
****************************************************
```

```
*** example (rounding computations):


cat -n moda_dur_3.c
/* entree */
double X;
__ASTREE_volatile_input((X [-186.,186.]));

/* sortie */
double RESULTAT;

void N()
{
   int tronc_entier;
    double  entree,diametre,min,rapport,troncature,plancher,multiple_inf,reste,reste
   int BPO;
   min = 0;
   diametre = 1.;

   /* au choix: nouvelle entree ou retroaction */
   if (BPO) entree = X;
   else     entree = RESULTAT;

   /* calcul du rapport de entree - min / diametre, puis de sa troncature */
   min = 0;
   diametre = 1.;
   rapport = (entree - min) / diametre;
   tronc_entier = (int) rapport;
   troncature = (double) tronc_entier;

   /* calcul de la valeur plancher de ce rapport */
   if (rapport<0) plancher = troncature - 1;
   else           plancher = troncature;

   /* calcul du reste de l'entree */
   reste = entree - (diametre * plancher);

```

```
    34      /* calcul du multiple inferieur a l'entree*/
    35      multiple_inf = entree - reste;
    36
    37      /* calcul du multiple superieur a l'entree*/
    38      multiple_sup = multiple_inf + diametre;
    39
    40
    41      /* calcul du multiple le plus proche */
    42      if (reste < 0) reste_abs = -reste;
    43      else           reste_abs = reste;
    44      if (reste_abs <= 0.5*diametre)  plus_proche = multiple_inf;
    45      else                             plus_proche = multiple_sup;
    46
    47
    48      /* resultat */
    49      RESULTAT = plus_proche;
    50        __ASTREE_log_vars((entree,RESULTAT;mod,inter));
    51    }
    52
    53
    54    void main()
    55    {
    56      while (1) {
    57        N();
    58          __ASTREE_wait_for_clock(());
    59      }
    60    }
```

```
*** static analysis by Astree (1 - WITHOUT abstract
     domain for troncations):


astree moda_dur_3.c --exec-fn main --no-nod |& egrep "(launched)|(WARN)"
/* Analyzer launched at 2006/ 7/12 06:48:07

moda_dur_3.c:24.17-30::[call#main@54:loop@56>=4:call#N@57:]: WARN: double->signed int conve
%

*** static analysis by Astree (2 - WITH abstract
     domain for troncations):


astree moda_dur_3.c --exec-fn main --mod |& egrep "(launched)|(WARN)"
/* Analyzer launched at 2006/ 7/12 06:48:09

%

*** troncation information derived by Astree:


astree moda_dur_3.c --exec-fn main --mod |& egrep --after-context 18  "(launched)|(WARN)|(c
   <interval: <> <entree in [-186.1, 186.1], RESULTAT in [-186.1, 186.1]> >
   <modulo:
     there exists an integer i in ((entree) - 0.)/1. + [-0.5;0.5]
such that: RESULTAT = 1.*.i + [-3.33733041202e-13;3.33733041202e-13]
>
   <modulo:
     tronc_entier = Arr_0(((entree) - 0.)/1. + [-0.;0.]) + [-0.;0.]
there exists an integer i in ((entree) - 0.)/1. + [-0.5;0.5]
such that: plus_proche = 1.*.i + [-3.33733041202e-13;3.33733041202e-13]
there exists an integer i in ((entree) - 0.)/1. + [-1.;8.26450019531e-14]
such that: reste=entree - 1.*i   + [-1.66111568944e-13;1.66111568944e-13]
there exists an integer i in ((entree) - 0.)/1. + [-1.;8.26450019531e-14]
such that: plancher = i + [-4.17443857259e-14;4.17443857259e-14]
troncature = Arr_0(((entree) - 0.)/1. + [-0.;0.]) + [-0.;0.]
rapport=((entree) - 0.)/1. + [-8.26450019531e-14;8.26450019531e-14]
```

```
there exists an integer i in ((entree) - 0.)/1. + [-0.5;0.5]
such that: RESULTAT = 1.*.i + [-3.33733041202e-13;3.33733041202e-13]
>
%


***********************************************
* Astree knows about synchronous programming... *
***********************************************



*** incorrect example:


cat -n clock-error.c
     1   /* clock-error.c */
     2   int R, T, n = 10;
     3   void main()
     4   { volatile int I;
     5     R = 0;
     6     while (1) {
     7       if (I)
     8         { R = R+1; }
     9       else
    10         { R = 0; }
    11       T = (R>=n);
    12   /*   __ASTREE_wait_for_clock(()); */
    13   }}


*** configuration file:

cat -n clock-error.config
     1   /* clock-error.config */
     2   __ASTREE_volatile_input((I [0,1]));


*** analysis of the incorrect example by Astree:


astree --exec-fn main --config-sem clock-error.config clock-error.c |& egrep "(launched)|(
/* Analyzer launched at 2006/ 7/12 06:48:14
clock-error.c:8.12-15::[call#main@3:loop@6>=4:]: WARN: signed int arithmetic range [-21474
%


*** correct example:

cat -n clock.c
     1   /* clock.c */
     2   int R, T, n = 10;
     3   void main()
     4   { volatile int I;
     5     R = 0;
     6     while (1) {
     7       if (I)
     8         { R = R+1; }
     9       else
    10         { R = 0; }
    11       T = (R>=n);
    12         __ASTREE_wait_for_clock(());
    13   }}


*** correction (difference with the incorrect program):
```

```
1c1
< /* clock-error.c */
---
> /* clock.c */
12c12
< /*   __ASTREE_wait_for_clock(()); */
---
>       __ASTREE_wait_for_clock(());


*** configuration file:

cat -n clock.config
     1	/* clock.config */
     2	__ASTREE_volatile_input((I  [0,1]));
     3	__ASTREE_max_clock((3600000));


*** analysis of the correct example by Astree:


astree --exec-fn main --config-sem clock.config clock.c |& egrep "(launched)|(WARN)"
/* Analyzer launched at 2006/ 7/12 06:48:17
%


************************************************
* Astree knows about control/command theory... *
************************************************



*** filter example:


cat -n filtre.c
     1	typedef enum {FALSE = 0, TRUE = 1} BOOLEAN;
     2	BOOLEAN INIT;
     3	float P, X;
     4	volatile float RANDOM_INPUT;
     5	__ASTREE_volatile_input((RANDOM_INPUT [-10.0,10.0]));
     6	
     7	void filtre2 () {
     8	   static float E[2], S[2];
     9	   if (INIT) {
    10	       S[0] = X;
    11	       P = X;
    12	       E[0] = X;
    13	   } else {
    14	        P = (((((0.4677826 * X) - (E[0] * 0.7700725)) + (E[1] * 0.4344376)) + (S[0
    15	   }
    16	   E[1] = E[0];
    17	   E[0] = X;
    18	   S[1] = S[0];
    19	   S[0] = P;
    20	}
    21	
    22	void main () {
    23	   X = RANDOM_INPUT;
    24	   INIT = TRUE;
    25	   while (TRUE) {
    26	     X = RANDOM_INPUT;
    27	     filtre2 ();
    28	     INIT = FALSE;
    29	   }
    30	}
```

```
*** static analysis by Astree (1 -- WITH 2nd order
     filter domain):


astree filtre.c --dump-invariants --exec-fn main |& egrep "(launched)|(WARN)|(P in)"
/* Analyzer launched at 2006/ 7/12 06:48:19

    X in [-10, 10], P in [-13.3881, 13.3881], RANDOM_INPUT in [-10, 10]> >
%

*** static analysis by Astree (2 -- WITHOUT 2nd order
     filter domain):

astree filtre.c --exec-fn main --no-filtre2 --dump-invariants |& egrep "(launched)|(WARN)|
/* Analyzer launched at 2006/ 7/12 06:48:21
filtre.c:14.6-114::[call#main@22:loop@25>=4:call#filtre2@27:]: WARN: double->float conversio
    X in [-10, 10], P in [-3.40282e+38, 3.40282e+38],

%

************************************************
* Astree can analyze low level memory operations *
************************************************



*** example 1 (pointer casts):


cat -n memcpy.c
     1   /* memcpy.c (polymorphic memcpy) */
     2
     3   /* byte per byte copy of src into dst */
     4   void memcpy(char* dst, const char* src, unsigned size)
     5   {
     6     int i;
     7     for (i=0;i<size;i++) dst[i] = src[i];
     8   }
     9
    10   void main()
    11   {
    12     float x = 10.0, y;
    13     int zero = 0;
    14     /* copy of x into y (well-typed) */
    15      memcpy(&y,&x,sizeof(y));
    16      __ASTREE_assert((y==10.0));
    17     /* copy of zero into y (not well-typed but allowed in C) */
    18      memcpy(&y,&zero,sizeof(y));
    19      __ASTREE_assert((y==0.0));
    20   }


*** static analysis by Astree:


astree --exec-fn main --unroll 5 memcpy.c |& egrep "(launched)|(WARN)"
/* Analyzer launched at 2006/ 7/12 06:48:23
%


*** example 2 (unions):


cat -n union.c
     1   /* union.c (union type) */
```

```
      2
      3   union {
      4     int type;
      5     struct { int type; int data; } A;
      6     struct { int type; char data[3]; } B;
      7   } u;
      8
      9   void main()
     10   {
     11     /* no assert failure */
     12     u.type = 12;
     13      __ASTREE_assert((u.A.type==12));
     14      __ASTREE_assert((u.B.type==12));
     15
     16     /* assert failure because the modification of u.B.data also modifies u.A.data *
     17     u.A.data = 0;
     18     u.B.data[0] = 12;
     19      __ASTREE_assert((u.A.data==0));
     20   }


*** static analysis by Astree:


astree --exec-fn main --full-memory-model union.c |& egrep "(launched)|(WARN)"
/* Analyzer launched at 2006/ 7/12 06:48:25
union.c:19.19-30::[call#main@9:]: WARN: assert failure
%


*********************************************
* Astree has a graphic interface under X11... *
*********************************************

tcsh: /bin/rm: No match.


*** static analysis by Astree


astree filtre.c --dump-invariants --exec-fn main --export-invariant-stat --export-file fil

%

*** visualization of the results:


visu --text-size 14 --text-font CMTT filtre.inv >& /dev/null &
[1] 13075
%

*** (scaling up with GTK+ (library to build graphical
      user interfaces (GUIs) originally for X Window)!)



********************************************
*** The end, thank you for your attention ***
********************************************

Publications/2006-Dagstuhl-06281/Cousot-Dagstuhl-06281_demonstration % ^Dexit

Script done on Wed Jul 12 08:48:32 2006
```