Some Notes on cdd

Description from the website, www.ifor.math.ethz.ch/~fukuda/cdd_home/cdd.html:

> The program cdd+ is a C++ implementation of the Double Description Method of Motzkin et al. for generating all vertices (i.e. extreme points) and extreme rays of a general convex polyhedron in $\mathbf{R}^d$ given by a system of linear inequalities:
>
> $$P = \{x : Ax \leq b\}$$
>
> where A is an $m \times d$ real matrix and b is a real m dimensional vector. The program can be used for the reverse operation (i.e. convex hull computation) if one run cdd with "hull" option. This means that one can move back and forth between an inequality representation and a generator (i.e. vertex and ray) representation of a polyhedron with cdd+. Also, cdd+ can solve a linear programming problem, i.e. a problem of maximizing and minimizing a linear function over P.
>
> The program cdd+ is a C++ version of the ANSI C program cdd basically for the same purpose. The main difference is that it can be compiled for both rational (exact) arithmetic and floating point arithmetic. (Note that cdd runs on floating arithmetic only.)

If you have an account on UK's Mathematical Sciences system, you can open a unix window by clicking "Start,", then "All Programs," then "Telnet T." Login and then run cdd from the unix prompt by typing the command

~lee/cdd/new/cdd-061/cdd

This is probably not the latest version of cdd, but I have not installed a later version yet.

To give you an example of using the program, first prepare the following file, calling it test1.ext:

```
begin
7 4 integer
1 0 0 0
1 1 0 0
1 0 1 0
1 0 0 1
1 1 1 0
1 1 0 1
1 0 1 1
end
hull
incidence
```

Each of the seven rows in the matrix gives the coordinates of a point in $\mathbf{R}^3$. The first coordinate is always a "1" for points, and a "0" for rays.

Supply this file as the input to cdd, typing

~lee/cdd/new/cdd-061/cdd test1.ext.

Then look at the output files that are created, test1.ine and test1.icd. Here is test1.ine:

```
* cdd: Double Description Method C-Code:Version 0.61 (December 1, 1997)
* Copyright (C) 1996, Komei Fukuda, fukuda@ifor.math.ethz.ch
*Input File:test1.ext   (   7  x    4)
*HyperplaneOrder: LexMin
*Degeneracy preknowledge for computation: None (possible degeneracy)
*Hull computation is chosen.
*Computation starts     at Wed Aug 17 14:10:04 2005
*            terminates at Wed Aug 17 14:10:04 2005
*Total processor time = 0 seconds
*                     = 0 hour 0 min 0 sec
*Since hull computation is chosen, the output is a minimal inequality
system
*FINAL RESULT:
*Number of Facets =       7
H-representation
```

```
begin
        7      4    real
  0  0  1  0
  1  0  0 -1
  0  1  0  0
  1  0 -1  0
  2 -1 -1 -1
  1 -1  0  0
  0  0  0  1
end
```

Each of the seven rows in the matrix provides an linear inequality in $\mathbf{R}^3$. For example, `2 -1 -1 -1` is to be interpreted as $2 - x - y - z \geq 0$.

Here is the file test1.icd:

```
*Incidences of output(=facets) and input (=points)
*   for each output, #incidence and the set of points lying on it
*   or its complement with its cardinality with minus sign
*cdd input file : test1.ext   (   7  x    4)
*cdd output file: test1.ine
begin
       7    7    7
 -3 : 3 5 7
 3 : 4 6 7
 -3 : 2 5 6
 3 : 3 5 7
 3 : 5 6 7
 3 : 2 5 6
 -3 : 4 6 7
end
```

Each row corresponds to an inequality, with the list of points lying on it, or the complement of the list of points lying on it (indicated with a minus sign). So for example, from the first row of the matrix we see that the list of points not lying on the first inequality is 3,5,7, so the points lying on the first inequality are 1,2,4,6.

You can now copy the file test1.ine as test2.ine and provide it as input to cdd to convert from inequalities back to vertices—try it.