

Gareth Evans | IPM 4097 Project | 2001-2002

Appendix C: Program Output

The following is the log file of a terminal session conducted on *April 30th, 2002* to demonstrate the **kba** and **kbe** programs in use. Comments in the transcript are shown in *red*.

BEGIN TRANSCRIPT

CONNECT TO publix.bangor.ac.uk USING TELNET

```
*****
* University of Wales, Bangor | Prifysgol Cymru, Bangor *
*                               |                               *
* If you have not been given  | Os nad oes gennych ganiatad i*
* permission to use this machine | ddefnyddio'r peiriant yma *
* then do NOT attempt to login. | peidiwch a ceisio gwneud. *
*                               | operator@bangor.ac.uk *
*****
Connected from : host213-122-0-84.in-addr.btopenworld.com
```

SunOS 5.8

login: mau402

Password:

Last login: Sat Apr 27 20:08:54 from host213-122-0-8
Sun Microsystems Inc. SunOS 5.8 Generic February 2000

```
-----
If you have any problems using this system or software on this system
please send E-mail to:
                                helpdesk
-----
explaining in sufficient detail what you require help with.
-----
```

Good Evening! There are currently 16 user(s) running 12 programs.

*LOG IN TO 'DEMO' IN ORDER TO ACCESS THE PROGRAM FILES IN THE DIRECTORY
/mw3/usr/lib/freemon2/gareth/kbia*

publix:/homedir/mau402>rlogin demo

Password:

Last login: Sat Apr 27 20:07:28 from publix
Copyright (c) 1980, 1983, 1986, 1988, 1990, 1991, 1993, 1994
The Regents of the University of California. All rights reserved.

FreeBSD 3.3-RELEASE (MATH) #0: Wed Oct 27 18:20:43 BST 1999

Croeso i'r Ysgol Mathemateg, Prifysgol Cymru Bangor

demo:/homedir/mau402> cd /mw3/usr/lib/freemon2/gareth/kbia

demo:gareth/kbia> ls -l

```
total 288
-rw-r--r-- 1 mau402 ma      66 Apr 30 13:21 a4.in
-rw-r--r-- 1 mau402 ma      72 Apr 30 13:21 a6.in
-rw-r--r-- 1 mau402 ma      77 Apr 30 13:21 adata_c2.in
-rw-r--r-- 1 mau402 ma      13 Apr 30 13:21 adata_s3.in
-rw-r--r-- 1 mau402 ma     113 Apr 30 13:22 adata_s3s4.in
-rw-r--r-- 1 mau402 ma      33 Apr 30 13:22 c2.in
-rw-r--r-- 1 mau402 ma      33 Apr 30 13:22 c3.in
-rw-r--r-- 1 mau402 ma      11 Apr 30 13:22 edata_c2c3.in
-rw-r--r-- 1 mau402 ma      33 Apr 30 13:22 edata_s3s4.in
-rw-r--r-- 1 mau402 ma      33 Apr 30 13:22 edata_s3s4kerI.in
-rw-r--r-- 1 mau402 ma      15 Apr 30 13:22 edata_s3s4kerc3.in
-rw-r--r-- 1 mau402 ma       9 Apr 30 13:22 edata_s3s4kers3.in
-rw-r--r-- 1 mau402 ma    27422 Apr 30 13:22 examples.txt
-rw-r--r-- 1 mau402 ma       76 Apr 30 13:22 ida.in
-rw-r--r-- 1 mau402 ma       78 Apr 30 13:22 idb.in
-rwxr-xr-x 1 mau402 ma   108513 Apr 30 13:22 kba*
-rw-r--r-- 1 mau402 ma     1702 Apr 30 13:22 kbahelp.txt
-rwxr-xr-x 1 mau402 ma   108953 Apr 30 13:22 kbe*
-rw-r--r-- 1 mau402 ma     1346 Apr 30 13:22 kbehelp.txt
-rw-r--r-- 1 mau402 ma       79 Apr 30 13:22 s3.in
-rw-r--r-- 1 mau402 ma       79 Apr 30 13:23 s4.in
-rw-r--r-- 1 mau402 ma       12 Apr 30 13:23 word.in
```

LET US NOW LOOK AT THE HELP FILES FOR THE TWO PROGRAMS...

HELP FILE FOR THE 'kba' PROGRAM:

demo:gareth/kbia>more kbahelp.txt

```
*****
* HOW TO USE THE KBA PROGRAM -- QUICK GUIDE *
*****
```

The standard way to run the program is as follows:
on the command line, type in

kba FILE1 (FILE2) (FILE3)
 e.g. 'kba s4.in'
 (Normal Knuth-Bendix on s4.in)
 or 'kba s4.in adata_s3s4.in'
 (Induced Action Knuth-Bendix)
 or 'kba s4.in adata_s3s4.in word.in'
 (Induced Action Knuth-Bendix with Example)

The three input files have the following format:

FILE 1:
 Standard Presentation for a monoid.

Example of Format:

```
C; D; c; d;
Cc ; ;
cC ; ;
Dd ; ;
dD ; ;
C^4 ; ;
D^3 ; ;
(C*D)^2 ; ;
```

Notes: First line = list of generators;
 Remaining lines = relators.

FILE 2: (OPTIONAL)
 Line 1: Number of Generators in M, terminated by a comma.
 Line 2: Images of the M generators separated by semicolons
 (in the order that the generators appear
 in the presentation file for M).

Remaining Lines: The M-action on each generator of M
 (again in the order that the generators
 appear in the presentation file for M).
 For example, if $|X| = 3$, then the M-action on the
 generator A given by $1^A = 2$, $2^A = 3$ and $3^A = 1$
 is written in the input file as 2, 3, 1,

Example of Format:

```
2,
D^2; C^3D*C^2;
2, 3, 1,
1, 3, 2,
```

Note: The numbers are terminated by spaces
 and there should be NO SPACES at the end of each line.

FILE 3 (OPTIONAL):
 Should contain a single word (x, n) to simplify using the
 complete rewrite systems computed in the program.

Example of Format:

```
2, D*C^2*D;
```

```
*****
* See the USER MANUAL for more information on how to use the program *
*****
```

HELP FILE FOR THE 'kbe' PROGRAM:

```
demo:gareth/kbia>more kbehelp.txt
*****
* HOW TO USE THE KBE PROGRAM -- QUICK GUIDE *
*****
```

The standard way to run the program is as follows:
 on the command line, type in

```
kbe FILE1 (FILE2) (FILE3) (FILE4)
```

e.g. 'kbe s4.in'
 (Normal Knuth-Bendix on s4.in)
 or 'kbe s3.in s4.in edata_s3s4.in'
 (Induced Action Knuth-Bendix)
 or 'kbe s3.in s4.in edata_s3s4.in word.in'
 (Induced Action Knuth-Bendix with Example)

The four input files have the following format:

FILE 1 and FILE 2:
 Standard Presentation for a monoid.

Example of Format:

```
C; D; c; d;
Cc ; ;
cC ; ;
Dd ; ;
dD ; ;
C^4 ; ;
D^3 ; ;
(C*D)^2 ; ;
```

Notes: First line = list of generators;
Remaining lines = relators.

FILE 3:

Line 1: Images of the M generators separated by semicolons
(in the order that the generators appear
in the presentation file for M).

Example of Format:

```
D^2; C^3*D*C^2; d^2; c^2*d*c^3;
```

Note: There should be NO SPACES at the end of the line.

FILE 4:

Should contain a single word (x, n) to simplify using the
complete rewrite systems computed in the program.

Example of Format:

```
2, D*C^2*D;
```

```
*****  
* See the USER MANUAL for more information on how to use the program *  
*****
```

LET US NOW LOOK AT SOME OF THE INPUT FILES THAT WE ARE GOING TO USE...

PRESENTATION FILE FOR THE SYMMETRIC GROUP S(3):

```
demo:gareth/kbia>more s3.in
```

```
A; B; a; b;
A*a ; ;
a*A ; ;
B*b ; ;
b*B ; ;
A^3 ; ;
B^2 ; ;
(A*B)^2 ; ;
```

PRESENTATION FILE FOR THE SYMMETRIC GROUP S(4):

```
demo:gareth/kbia>more s4.in
```

```
C; D; c; d;
C*c ; ;
c*C ; ;
D*d ; ;
d*D ; ;
C^4 ; ;
D^3 ; ;
(C*D)^2 ; ;
```

PRESENTATION FILE FOR THE CYCLIC GROUP C(2):

```
demo:gareth/kbia>more c2.in
```

```
A; a;
A*a ; ;
a*A ; ;
A^2 ; ;
```

PRESENTATION FILE FOR THE CYCLIC GROUP C(3):

```
demo:gareth/kbia>more c3.in
```

```
B; b;
B*b ; ;
b*B ; ;
B^3 ; ;
```

A FILE TO STORE AN ARBITRARY WORD FROM THE SET $X^{<N>}$ FOR REDUCTION USING A COMPLETE REWRITE SYSTEM:

```
demo:gareth/kbia>more word.in
```

```
2, (C*D)^5;
```

INFORMATION FILE (TO BE USED IN THE 'kba' PROGRAM)

FOR THE INDUCED MONOID ACTION ASSOCIATED WITH $M = S(3)$, $N = S(4)$, $f: S(3) \rightarrow S(4)$ AND $X = C(2)$:

```
demo:gareth/kbia>more adata_c2.in
```

```
4,  
D^2; C^3*D*C^2; d^2; c^2*d*c^3;  
2, 3, 1,  
1, 3, 2,  
3, 1, 2,  
1, 3, 2,
```

INFORMATION FILE (TO BE USED IN THE 'kba' PROGRAM)
 FOR THE INDUCED MONOID ACTION ASSOCIATED WITH $M = S(3)$, $N = S(4)$, $f: S(3) \rightarrow S(4)$ AND $X = S(3)$:
 demo:gareth/kbia>more adata_s3s4.in
 4,
 D^2; C^3*D*C^2; d^2; c^2*d*c^3;
 2, 4, 6, 1, 3, 5,
 3, 5, 1, 6, 2, 4,
 4, 1, 5, 2, 6, 3,
 3, 5, 1, 6, 2, 4,

INFORMATION FILE (TO BE USED IN THE 'kbe' PROGRAM)
 FOR THE INDUCED MONOID ACTION ASSOCIATED WITH $M = S(3)$, $N = S(4)$, $f: S(3) \rightarrow S(4)$ AND $X = S(3)$:
 demo:gareth/kbia>more edata_s3s4.in
 D^2; C^3*D*C^2; d^2; c^2*d*c^3;

INFORMATION FILE (TO BE USED IN THE 'kbe' PROGRAM)
 FOR THE INDUCED MONOID ACTION ASSOCIATED WITH $M = C(2)$, $N = C(3)$, $f: C(2) \rightarrow C(3)$ AND $X = C(2)$:
 demo:gareth/kbia>more edata_c2c3.in
 B^3; b^3;

WE NOW COME TO USING THE PROGRAMS....

FIRST, WE SHALL APPLY THE 'NORMAL' KNUTH-BENDIX CRITICAL PAIRS COMPLETION ALGORITHM
 TO THE SYMMETRIC GROUP $S(3)$:

demo:gareth/kbia> kba s3.in

Data read in. Now processing...

Generating set for monoid $N =$

A
 B
 a
 b
 [4 generator(s)]

N is presented as follows:

(A a -> 1)
 (a A -> 1)
 (B b -> 1)
 (b B -> 1)
 (A^3 -> 1)
 (B^2 -> 1)
 (A B A B -> 1)
 [7 rule(s)]

Now computing a complete rewrite system for N

In fMonKnuthBendix using RemDups:

Pass number 1:
 8 critical pairs added; 7 rules in the reduced set.

Pass number 2:
 8 critical pairs added; 10 rules in the reduced set.

Pass number 3:
 0 critical pairs added; 10 rules in the reduced set.

Number of passes made = 3.

...Complete rewrite system for N computed.

This is the Complete rewrite system for N :

(B a -> A B)
 (a^2 -> A)
 (a B -> B A)
 (B^2 -> 1)
 (a A -> 1)
 (A a -> 1)
 (A^2 -> a)
 (B A B -> a)
 (A B A -> b)
 (b -> B)
 [10 rule(s)]

Elements of the monoid N :

1
 A
 B
 a
 A B
 B A
 [6 element(s)]

NEXT, WE SHALL APPLY THE 'NORMAL' KNUTH-BENDIX CRITICAL PAIRS COMPLETION ALGORITHM TO THE CYCLIC GROUP C(3):

demo:gareth/kbia> kbe c3.in

Data read in. Now processing...

Generating set for monoid M =
B
b
[2 generator(s)]

M is presented as follows:
(B b -> 1)
(b B -> 1)
(B^3 -> 1)
[3 rule(s)]

Now computing a complete rewrite system for M.....

In fMonKnuthBendix using RemDups:

Pass number 1:
2 critical pairs added; 3 rules in the reduced set.

Pass number 2:
2 critical pairs added; 4 rules in the reduced set.

Pass number 3:
0 critical pairs added; 4 rules in the reduced set.

Number of passes made = 3.

...Complete rewrite system for M computed.

This is the Complete rewrite system for M:
(b^2 -> B)
(b B -> 1)
(B b -> 1)
(B^2 -> b)
[4 rule(s)]

Elements of the monoid M:
1
B
b
[3 element(s)]

WE SHALL NOW APPLY THE MODIFIED 'INDUCED ACTION' KNUTH-BENDIX CRITICAL PAIRS COMPLETION ALGORITHM IN THE SITUATION WHERE M = S(3), N = S(4), f: S(3) -> S(4) (AS SPECIFIED IN adata_c2.in) AND X = C(2):

demo:gareth/kbia> kba s4.in adata_c2.in

Data read in. Now processing...

Generating set for monoid N =
C
D
c
d
[4 generator(s)]

N is presented as follows:
(C c -> 1)
(c C -> 1)
(D d -> 1)
(d D -> 1)
(C^4 -> 1)
(D^3 -> 1)
(C D C D -> 1)
[7 rule(s)]

Now computing a complete rewrite system for N.....

In fMonKnuthBendix using RemDups:

Pass number 1:
8 critical pairs added; 8 rules in the reduced set.

Pass number 2:
12 critical pairs added; 16 rules in the reduced set.

Pass number 3:
5 critical pairs added; 18 rules in the reduced set.

Pass number 4:
5 critical pairs added; 21 rules in the reduced set.

Pass number 5:
0 critical pairs added; 21 rules in the reduced set.

Number of passes made = 5.

...Complete rewrite system for N computed.

These are the Type 1 rules in the Complete Rewrite System:

```
(c D c D -> D C^2 d)
(D c D c -> C D c D)
(D C^2 d C -> C D c D c)
(D C^2 D -> c D c)
(d C d -> D c D)
(D C d -> c D)
(d C D -> D c)
(d C^2 -> C D c)
(c D C -> C^2 d)
(d c -> C D)
(c^2 -> C^2)
(c d -> D C)
(d^2 -> D)
(d D -> 1)
(D d -> 1)
(c C -> 1)
(C c -> 1)
(C^3 -> c)
(D C D -> c)
(C D C -> d)
(D^2 -> d)
[21 rule(s)]
```

Continuing to process the data...

The number of generators of M is 4.

Images of M generators:

```
D^2
C^3 D C^2
d^2
c^2 d c^3
[4 image(s)]
```

Type 2 rules for the initial rewrite system:

```
(1, d) -> (2, 1)
(2, d) -> (3, 1)
(3, d) -> (1, 1)
(1, C^2 d C) -> (1, 1)
(2, C^2 d C) -> (3, 1)
(3, C^2 d C) -> (2, 1)
(1, D) -> (3, 1)
(2, D) -> (1, 1)
(3, D) -> (2, 1)
(1, C^2 d C) -> (1, 1)
(2, C^2 d C) -> (3, 1)
(3, C^2 d C) -> (2, 1)
[12 rule(s)]
```

Now computing the complete rewrite system for elements of the form (x, n) :

Checking the initial rewrite system...
...Initial rewrite system checked.

Iteration 1...

```
...9 elements to begin with
...0 elements added of Type 2 vs. Type 2
...20 elements added of Type 1 vs. Type 2
...21 elements in the reduced set
```

Iteration 2...

```
...21 elements to begin with
...0 elements added of Type 2 vs. Type 2
...0 elements added of Type 1 vs. Type 2
...21 elements in the reduced set
```

...Complete rewrite system computed for elements of the form (x, n) .

The following is the Complete Rewrite System:

-----TYPE 1 RULES-----

```
(c D c D -> D C^2 d)
(D c D c -> C D c D)
(D C^2 d C -> C D c D c)
(D C^2 D -> c D c)
(d C d -> D c D)
(D C d -> c D)
(d C D -> D c)
(d C^2 -> C D c)
(c D C -> C^2 d)
(d c -> C D)
(c^2 -> C^2)
(c d -> D C)
(d^2 -> D)
(d D -> 1)
(D d -> 1)
(c C -> 1)
(C c -> 1)
(C^3 -> c)
(D C D -> c)
(C D C -> d)
(D^2 -> d)
[21 rule(s)]
```

-----TYPE 2 RULES-----

```
(2, C d) -> (2, C^2)
(3, C^2 d) -> (2, c)
(2, C^2 D) -> (2, C)
(1, C d) -> (3, C^2)
(2, C^2 d) -> (3, c)
(1, d) -> (2, 1)
(2, d) -> (3, 1)
(3, d) -> (1, 1)
(1, D) -> (3, 1)
(2, D) -> (1, 1)
(3, D) -> (2, 1)
(3, c D) -> (2, C^2)
(2, C D) -> (3, c)
(1, C D) -> (2, c)
(1, c D) -> (1, C^2)
(3, C D) -> (1, c)
(2, c D) -> (3, C^2)
(3, C^2 D) -> (1, C)
(1, C^2 d) -> (1, c)
(3, C d) -> (1, C^2)
(1, C^2 D) -> (3, C)
[21 rule(s)]
```

-----ELEMENTS-----

```
[1, 1]
[2, 1]
[3, 1]
[1, C]
[1, c]
[2, C]
[2, c]
[3, C]
[3, c]
[1, C^2]
[3, C^2]
[2, C^2]
[12 element(s)]
```

**LET US NOW APPLY THE MODIFIED 'INDUCED ACTION' KNUTH-BENDIX CRITICAL PAIRS
COMPLETION ALGORITHM IN THE SITUATION WHERE $M = S(3)$, $N = S(4)$,
 $f: S(3) \rightarrow S(4)$ (AS SPECIFIED IN `adata_s3s4.in`) AND $X = S(3)$:**
demo:gareth/kbia> kba s4.in adata_s3s4.in

Data read in. Now processing...

Generating set for monoid N =

```
C
D
c
d
[4 generator(s)]
```

N is presented as follows:

```
(C c -> 1)
(c C -> 1)
(D d -> 1)
(d D -> 1)
(C^4 -> 1)
(D^3 -> 1)
(C D C D -> 1)
[7 rule(s)]
```

Now computing a complete rewrite system for N....

In fMonKnuthBendix using RemDups:

Pass number 1:
8 critical pairs added; 8 rules in the reduced set.

Pass number 2:
12 critical pairs added; 16 rules in the reduced set.

Pass number 3:
5 critical pairs added; 18 rules in the reduced set.

Pass number 4:
5 critical pairs added; 21 rules in the reduced set.

Pass number 5:
0 critical pairs added; 21 rules in the reduced set.

Number of passes made = 5.

...Complete rewrite system for N computed.

These are the Type 1 rules in the Complete Rewrite System:

```
(c D c D -> D C^2 d)
(D c D c -> C D c D)
(D C^2 d C -> C D c D c)
(D C^2 D -> c D c)
(d C d -> D c D)
(D C d -> c D)
(d C D -> D c)
(d C^2 -> C D c)
(c D C -> C^2 d)
(d c -> C D)
(c^2 -> C^2)
(c d -> D C)
(d^2 -> D)
(d D -> 1)
(D d -> 1)
(c C -> 1)
(C c -> 1)
(C^3 -> c)
(D C D -> c)
(C D C -> d)
(D^2 -> d)
[21 rule(s)]
```

Continuing to process the data...

The number of generators of M is 4.

Images of M generators:

```
D^2
C^3 D C^2
d^2
c^2 d c^3
[4 image(s)]
```

Type 2 rules for the initial rewrite system:

```
(1, d) -> (2, 1)
(2, d) -> (4, 1)
(3, d) -> (6, 1)
(4, d) -> (1, 1)
(5, d) -> (3, 1)
(6, d) -> (5, 1)
(1, C^2 d C) -> (3, 1)
(2, C^2 d C) -> (5, 1)
(3, C^2 d C) -> (1, 1)
(4, C^2 d C) -> (6, 1)
(5, C^2 d C) -> (2, 1)
(6, C^2 d C) -> (4, 1)
(1, D) -> (4, 1)
(2, D) -> (1, 1)
(3, D) -> (5, 1)
(4, D) -> (2, 1)
(5, D) -> (6, 1)
(6, D) -> (3, 1)
(1, C^2 d C) -> (3, 1)
(2, C^2 d C) -> (5, 1)
(3, C^2 d C) -> (1, 1)
(4, C^2 d C) -> (6, 1)
(5, C^2 d C) -> (2, 1)
(6, C^2 d C) -> (4, 1)
[24 rule(s)]
```

Now computing the complete rewrite system for elements of the form (x, n) :

Checking the initial rewrite system...
...Initial rewrite system checked.

Iteration 1...

...18 elements to begin with
...0 elements added of Type 2 vs. Type 2
...39 elements added of Type 1 vs. Type 2
...42 elements in the reduced set

Iteration 2...

...42 elements to begin with
...0 elements added of Type 2 vs. Type 2
...0 elements added of Type 1 vs. Type 2
...42 elements in the reduced set

...Complete rewrite system computed for elements of the form (x, n) .

The following is the Complete Rewrite System:

-----TYPE 1 RULES-----

(c D c D -> D C^2 d)
(D c D c -> C D c D)
(D C^2 d C -> C D c D c)
(D C^2 D -> c D c)
(d C d -> D c D)
(D C d -> c D)
(d C D -> D c)
(d C^2 -> C D c)
(c D C -> C^2 d)
(d c -> C D)
(c^2 -> C^2)
(c d -> D C)
(d^2 -> D)
(d D -> 1)
(D d -> 1)
(c C -> 1)
(C c -> 1)
(C^3 -> c)
(D C D -> c)
(C D C -> d)
(D^2 -> d)
[21 rule(s)]

-----TYPE 2 RULES-----

(3, C d) -> (4, C^2)
(6, C^2 d) -> (4, c)
(6, C d) -> (2, C^2)
(5, C^2 d) -> (2, c)
(6, C^2 D) -> (2, C)
(2, C d) -> (6, C^2)
(4, C^2 d) -> (6, c)
(3, C^2 D) -> (4, C)
(5, C d) -> (1, C^2)
(3, C^2 d) -> (1, c)
(5, C^2 D) -> (1, C)
(1, d) -> (2, 1)
(2, d) -> (4, 1)
(3, d) -> (6, 1)
(4, d) -> (1, 1)
(5, d) -> (3, 1)
(6, d) -> (5, 1)
(1, D) -> (4, 1)
(2, D) -> (1, 1)
(3, D) -> (5, 1)
(4, D) -> (2, 1)
(5, D) -> (6, 1)
(6, D) -> (3, 1)
(4, c D) -> (6, C^2)
(2, C D) -> (4, c)
(1, C D) -> (2, c)
(1, c D) -> (3, C^2)
(4, C D) -> (1, c)
(5, c D) -> (2, C^2)
(6, C D) -> (5, c)
(3, C D) -> (6, c)
(2, c D) -> (5, C^2)
(6, c D) -> (4, C^2)
(5, C D) -> (3, c)
(3, c D) -> (1, C^2)
(4, C^2 D) -> (3, C)
(1, C^2 d) -> (3, c)
(4, C d) -> (3, C^2)

```
(1, C^2 D) -> (5, C)
(2, C^2 d) -> (5, c)
(1, C d) -> (5, C^2)
(2, C^2 D) -> (6, C)
[42 rule(s)]
```

-----ELEMENTS-----

```
[1, 1]
[2, 1]
[3, 1]
[4, 1]
[5, 1]
[6, 1]
[1, C]
[1, c]
[2, C]
[2, c]
[3, C]
[3, c]
[4, C]
[4, c]
[5, C]
[5, c]
[6, C]
[6, c]
[1, C^2]
[5, C^2]
[3, C^2]
[2, C^2]
[6, C^2]
[4, C^2]
[24 element(s)]
```

LET US NOW APPLY THE MODIFIED 'INDUCED ACTION' KNUTH-BENDIX CRITICAL PAIRS COMPLETION ALGORITHM IN THE SITUATION WHERE $M = S(3)$, $N = S(4)$, $f: S(3) \rightarrow S(4)$ (AS SPECIFIED IN edata_s3s4.in) AND $X = S(3)$, NOTING THAT WE REDUCE THE WORD IN THE FILE word.in AT THE END OF THE OUTPUT:
demo:gareth/kbia> kbe s3.in s4.in edata_s3s4.in word.in

Data read in. Now processing...

Generating set for monoid $M =$

```
A
B
a
b
[4 generator(s)]
```

M is presented as follows:

```
(A a -> 1)
(a A -> 1)
(B b -> 1)
(b B -> 1)
(A^3 -> 1)
(B^2 -> 1)
(A B A B -> 1)
[7 rule(s)]
```

Now computing a complete rewrite system for M

In fMonKnuthBendix using RemDups:

Pass number 1:
8 critical pairs added; 7 rules in the reduced set.

Pass number 2:
8 critical pairs added; 10 rules in the reduced set.

Pass number 3:
0 critical pairs added; 10 rules in the reduced set.

Number of passes made = 3.

...Complete rewrite system for M computed.

This is the Complete rewrite system for M :

```
(B a -> A B)
(a^2 -> A)
(a B -> B A)
(B^2 -> 1)
(a A -> 1)
(A a -> 1)
(A^2 -> a)
(B A B -> a)
(A B A -> b)
(b -> B)
[10 rule(s)]
```

Elements of the monoid M:

1
A
B
a
A B
B A
[6 element(s)]

Continuing to process the data...

Generating set for monoid N =

C
D
c
d
[4 generator(s)]

N is presented as follows:

(C c -> 1)
(c C -> 1)
(D d -> 1)
(d D -> 1)
(C^4 -> 1)
(D^3 -> 1)
(C D C D -> 1)
[7 rule(s)]

Now computing a complete rewrite system for N.....

In fMonKnuthBendix using RemDups:

Pass number 1:
8 critical pairs added; 8 rules in the reduced set.

Pass number 2:
12 critical pairs added; 16 rules in the reduced set.

Pass number 3:
5 critical pairs added; 18 rules in the reduced set.

Pass number 4:
5 critical pairs added; 21 rules in the reduced set.

Pass number 5:
0 critical pairs added; 21 rules in the reduced set.

Number of passes made = 5.

...Complete rewrite system for N computed.

This is the complete rewrite system for N:

(c D c D -> D C^2 d)
(D c D c -> C D c D)
(D C^2 d C -> C D c D c)
(D C^2 D -> c D c)
(d C d -> D c D)
(D C d -> c D)
(d C D -> D c)
(d C^2 -> C D c)
(c D C -> C^2 d)
(d c -> C D)
(c^2 -> C^2)
(c d -> D C)
(d^2 -> D)
(d D -> 1)
(D d -> 1)
(c C -> 1)
(C c -> 1)
(C^3 -> c)
(D C D -> c)
(C D C -> d)
(D^2 -> d)
[21 rule(s)]

Continuing to process the data...

The number of generators in M is 4.

Images of M generators:

D^2
C^3 D C^2
d^2
c^2 d c^3
[4 image(s)]

Type 2 rules for the initial rewrite system:

```
(1, d) -> (2, 1)
(2, d) -> (4, 1)
(3, d) -> (6, 1)
(4, d) -> (1, 1)
(5, d) -> (3, 1)
(6, d) -> (5, 1)
(1, C^2 d C) -> (3, 1)
(2, C^2 d C) -> (5, 1)
(3, C^2 d C) -> (1, 1)
(4, C^2 d C) -> (6, 1)
(5, C^2 d C) -> (2, 1)
(6, C^2 d C) -> (4, 1)
(1, D) -> (4, 1)
(2, D) -> (1, 1)
(3, D) -> (5, 1)
(4, D) -> (2, 1)
(5, D) -> (6, 1)
(6, D) -> (3, 1)
(1, C^2 d C) -> (3, 1)
(2, C^2 d C) -> (5, 1)
(3, C^2 d C) -> (1, 1)
(4, C^2 d C) -> (6, 1)
(5, C^2 d C) -> (2, 1)
(6, C^2 d C) -> (4, 1)
[24 rule(s)]
```

Now computing a complete rewrite system for elements of the form (x, n):

Checking the initial rewrite system...
...Initial rewrite system checked.

Iteration 1...

...18 elements to begin with
...0 elements added of Type 2 vs. Type 2
...39 elements added of Type 1 vs. Type 2
...42 elements in the reduced set

Iteration 2...

...42 elements to begin with
...0 elements added of Type 2 vs. Type 2
...0 elements added of Type 1 vs. Type 2
...42 elements in the reduced set

...Complete rewrite system computed for elements of the form (x, n).

The following is the Complete Rewrite System:

```
-----TYPE 1 RULES-----
(c D c D -> D C^2 d)
(D c D c -> C D c D)
(D C^2 d C -> C D c D c)
(D C^2 D -> c D c)
(d C d -> D c D)
(D C d -> c D)
(d C D -> D c)
(d C^2 -> C D c)
(c D C -> C^2 d)
(d c -> C D)
(c^2 -> C^2)
(c d -> D C)
(d^2 -> D)
(d D -> 1)
(D d -> 1)
(c C -> 1)
(C c -> 1)
(C^3 -> c)
(D C D -> c)
(C D C -> d)
(D^2 -> d)
[21 rule(s)]
```

```
-----TYPE 2 RULES-----
(3, C d) -> (4, C^2)
(6, C^2 d) -> (4, c)
(6, C d) -> (2, C^2)
(5, C^2 d) -> (2, c)
(6, C^2 D) -> (2, C)
(2, C d) -> (6, C^2)
(4, C^2 d) -> (6, c)
(3, C^2 D) -> (4, C)
(5, C d) -> (1, C^2)
(3, C^2 d) -> (1, c)
(5, C^2 D) -> (1, C)
```

```

(1, d) -> (2, 1)
(2, d) -> (4, 1)
(3, d) -> (6, 1)
(4, d) -> (1, 1)
(5, d) -> (3, 1)
(6, d) -> (5, 1)
(1, D) -> (4, 1)
(2, D) -> (1, 1)
(3, D) -> (5, 1)
(4, D) -> (2, 1)
(5, D) -> (6, 1)
(6, D) -> (3, 1)
(4, c D) -> (6, C^2)
(2, C D) -> (4, c)
(1, C D) -> (2, c)
(1, c D) -> (3, C^2)
(4, C D) -> (1, c)
(5, c D) -> (2, C^2)
(6, C D) -> (5, c)
(3, C D) -> (6, c)
(2, c D) -> (5, C^2)
(6, c D) -> (4, C^2)
(5, C D) -> (3, c)
(3, c D) -> (1, C^2)
(4, C^2 D) -> (3, C)
(1, C^2 d) -> (3, c)
(4, C d) -> (3, C^2)
(1, C^2 D) -> (5, C)
(2, C^2 d) -> (5, c)
(1, C d) -> (5, C^2)
(2, C^2 D) -> (6, C)
[42 rule(s)]

```

-----ELEMENTS-----

```

[1, 1]
[2, 1]
[3, 1]
[4, 1]
[5, 1]
[6, 1]
[1, C]
[1, c]
[2, C]
[2, c]
[3, C]
[3, c]
[4, C]
[4, c]
[5, C]
[5, c]
[6, C]
[6, c]
[1, C^2]
[5, C^2]
[3, C^2]
[2, C^2]
[6, C^2]
[4, C^2]
[24 element(s)]

```

The word (x, n) to reduce is as follows:
(2, C D C D C D C D C D)
Now Reducing.....

Iteration 1...
Element to be reduced: (2, C D C D C D C D C D)
FMon reduction: C D C D C D C D C D -> C D
New element to be reduced: (2, C D)
Looking for overlaps in Type 2 elements...
Overlap found in (2, C D), (4, c) of length 2

Iteration 2...
Element to be reduced: (4, c)
FMon reduction: c -> c
New element to be reduced: (4, c)
Looking for overlaps in Type 2 elements...

...Reduction Finished.
The normalised form of the word is as follows
(it belongs to the following equivalence class):
[4, c]

TO FINISH, LET US APPLY THE MODIFIED 'INDUCED ACTION' KNUTH-BENDIX CRITICAL PAIRS
COMPLETION ALGORITHM IN THE SITUATION WHERE $M = C(2)$, $N = C(3)$,
 $f: C(2) \rightarrow C(3)$ (AS SPECIFIED IN `edata_c2c3.in`) AND $X = C(2)$:
`demo:gareth/kbia> kbe c2.in c3.in edata_c2c3.in`

Data read in. Now processing...

Generating set for monoid $M =$

A
a
[2 generator(s)]

M is presented as follows:

(A a \rightarrow 1)
(a A \rightarrow 1)
(A² \rightarrow 1)
[3 rule(s)]

Now computing a complete rewrite system for M

In `fMonKnuthBendix` using `RemDups`:

Pass number 1:
2 critical pairs added; 2 rules in the reduced set.

Pass number 2:
0 critical pairs added; 2 rules in the reduced set.

Number of passes made = 2.

...Complete rewrite system for M computed.

This is the Complete rewrite system for M :

(a \rightarrow A)
(A² \rightarrow 1)
[2 rule(s)]

Elements of the monoid M :

1
A
[2 element(s)]

Continuing to process the data...

Generating set for monoid $N =$

B
b
[2 generator(s)]

N is presented as follows:

(B b \rightarrow 1)
(b B \rightarrow 1)
(B³ \rightarrow 1)
[3 rule(s)]

Now computing a complete rewrite system for N

In `fMonKnuthBendix` using `RemDups`:

Pass number 1:
2 critical pairs added; 3 rules in the reduced set.

Pass number 2:
2 critical pairs added; 4 rules in the reduced set.

Pass number 3:
0 critical pairs added; 4 rules in the reduced set.

Number of passes made = 3.

...Complete rewrite system for N computed.

This is the complete rewrite system for N :

(b² \rightarrow B)
(b B \rightarrow 1)
(B b \rightarrow 1)
(B² \rightarrow b)
[4 rule(s)]

Continuing to process the data...

The number of generators in M is 2.

Images of M generators:

B³
b³
[2 image(s)]

Type 2 rules for the initial rewrite system:

```
(1, 1) -> (2, 1)
(2, 1) -> (1, 1)
(1, 1) -> (2, 1)
(2, 1) -> (1, 1)
[4 rule(s)]
```

Now computing a complete rewrite system for elements of the form (x, n) :

Checking the initial rewrite system...

```
(1, 1), (2, 1) is incorrectly ordered - changing to (2, 1), (1, 1)...
...Initial rewrite system checked.
```

Iteration 1...

```
...1 elements to begin with
...0 elements added of Type 2 vs. Type 2
...0 elements added of Type 1 vs. Type 2
...1 elements in the reduced set
```

...Complete rewrite system computed for elements of the form (x, n) .

The following is the Complete Rewrite System:

-----TYPE 1 RULES-----

```
(b^2 -> B)
(b B -> 1)
(B b -> 1)
(B^2 -> b)
[4 rule(s)]
```

-----TYPE 2 RULES-----

```
(2, 1) -> (1, 1)
[1 rule(s)]
```

-----ELEMENTS-----

```
[1, 1]
[1, B]
[1, b]
[3 element(s)]
```

```
demo:gareth/kbia> exit
logout
Connection closed.
```

END OF TRANSCRIPT