ITC

A GAP4 Share Package

Version 1.1

by

Volkmar Felsch, Ludger Hippe, and Joachim Neubüser

Lehrstuhl D für Mathematik
RWTH Aachen

March 2001

Contents

1

What is ITC?

ITC stands for **Interactive Todd Coxeter**, it is a program that allows you to execute interactively single steps in an enumeration of the cosets of a subgroup of a finitely presented group using the graphics surface XGAP of GAP and thus to see in various windows exactly what is happening. In this chapter we will

– discuss the background situation that led to the concept of this program,

– very briefly point to literature about coset enumeration,

– talk about envisaged use of ITC, and

– explain some technical points.

1.1 Background

Coset Enumeration, in the sequel abbreviated as CE, also known as the Todd-Coxeter procedure, is not only the oldest but also one of the most often used tools in Computational Group Theory and also probably the most often implemented one.

An easy introduction to CE and some of its variants is given in [Neu82]. In his book [Sim94] Charles Sims puts CE into a much more general context based on the theory of automata. Both texts provide a fairly comprehensive guide to the extensive literature on CE. So we will assume in the sequel that the user of this manual understands the basic ideas of CE. We will use as far as possible the terminology of [Neu82].

The main GAP library does contain routines for executing coset enumerations (see Section 45.5 in the GAP reference manual). These routines are partly written in the GAP language but for efficiency also use GAP kernel functions written in C. Some of these programs are used by ITC.

The most efficient implementation of CE, allowing many "strategies", at present is the package ACE by George Havas and Colin Ramsey, which will also soon be available as a share package of GAP (see [Ram99]).

It should be understood that ITC is no competitor for ACE. ACE can successfully deal with coset enumerations with millions of cosets. On the other hand handling coset tables graphically on the screen, even with using scrolling, in practice limits the use of ITC to tables with a few thousand rows. For that reason also the speed of the single step in the enumeration does not matter as much and ITC therefore is not tuned for speed in the way ACE is.

1.2 Teaching CE

In texts on CE usually examples are demonstrated, and whenever one teaches the method in class, one will have to work through some examples. In printing as well as on the blackboard this suffers from a certain difficulty: In CE a number of tables, the "coset table", the "subgroup tables", and the "relation tables" are being changed all the time, one has to gain information from (the closing of rows of) subgroup and relation tables and has to transfer this information into all relevant places of all tables. Sometimes, when so-called "coincidences" occur, one even has to erase and modify already existing entries in tables. Describing such steps in some detail as well as the ever changing tables soon occupy many pages in printing. Doing them in front of a class one can pretty soon get confused, forgetting to make entries or changes etc.

So it is a natural demand to have a program that allows one to perform such steps in windows on the screen of a computer where such changes can be cleanly made, are transferred to all relevant places in all tables automatically, and are marked by colours so that one can really "play" through nontrivial examples. This is what ITC provides. So in a first place it can be considered as a teaching tool.

1.3 Experimenting with CE Strategies

However we hope that ITC can also be used for experiments that ultimately may lead to a better understanding of existing methods or even the design of new ones.

As explained in many papers about CE, the sequence in which steps are made has great influence on the efficiency of the methods. To be more exact: CE is basically a method of trial and error in which one defines cosets of a subgroup of a finitely presented group by constructing coset representatives as words in the generators of the group. But only the termination of the method (the closing of the tables) in the end guarantees that one has not made a mistake by defining more than one word representing the same coset. The discovery of such a mistake and its removal is what is called "handling a coincidence" and it is the number of occurrences of such mistakes (the number of coincidences encountered) that may vary immensely depending on the chosen sequence of definition of pretended representatives.

Therefore many different "strategies" for CE have been discussed in the literature, and programs such as ACE allow one to use a wide variety of such strategies. However in spite of very detailed case studies, in particular by George Havas, that lead to certain "rules of thumb", no strategy is uniformly the best one, a strategy that is good for one presentation may turn out to be much less good for another one (see for instance [CDHW73], [Hav91], [HR01], and [HR99]). The possibility to try one's own strategy stepwise interactively therefore can be used to get some more insight into CE which to some extent is an art rather than a science.

Already in the building of ITC a nice side product was obtained from such experimentation: It is important to understand that in many cases coincidences cannot be completely avoided for a CE to terminate. However when many coincidences have occurred, one is led to ask if there is not a shorter sequence of definitions leading to the closure of the tables. Starting from a given CE with coincidences involved, one may try to "prune" the definition sequence from definitions that in retrospect can be recognized as being redundant. Such a procedure has for instance been employed many years ago "by hand" by John Leech in a search for a short definition sequence for the Fibonacci Group $F(2,7)$ that we will take as one of our examples. A very nice discussion of this idea and its history is given in Margaret Edeson's Master's Thesis (see [Ede89]). Using ITC such "pruning" was first done interactively. This led to an algorithm that is described in the section "The Short-cut Method" (see Section 3.5). It is now part of ITC and can be called by `short-cut` (see 5.2.9). In fact with the above mentioned old challenge problem of $F(2,7)$, it beats the formerly best known definition sequence leading to closure of the tables.

Since ITC will store sequences of definitions we intend in the future to investigate the possibility to improve runs of a "Modified TC" (MTC) by using such "pruned" definition sequences.

1.4 How does it work?

ITC is written using the graphics facilities provided by the GAP4 Share Package XGAP. We refer to the XGAP manual for a general description of these.

The ITC Coset Table as well as the Relation Tables and Subgroup Tables can be depicted in windows, and definitions be made in these by mouse click. Moreover tables with the sequence of definitions, of pending coincidences, and of gaps of length 1 in Subgroup or Relation Tables can be inspected in further windows and be used for deciding about the next step. A number of menus allow certain actions to be called.

In addition to the Coset Table Window which plays a special role, three different kinds of windows are used in ITC:

Pull-down menus

are used with the "top buttons" of the Coset Table (see 5.1). Clicking one of these will pop up a menu from which you can select actions of ITC. You do this by holding the left mouse button while moving the pointer to the respective entry in the menu and releasing the mouse button there. Then the respective action of the program will be executed while the menu vanishes.

Query windows

(sometimes also called transient windows) are used when the user has to specify additonal (in most cases numerical) data for an action to be performed. E.g. a user wanting to fill rows has to specify in such a query window which rows shall be filled. Note that no further action can be taken while such a query window is open. All other possibilities to operate ITC are disabled until the user has made the needed decision and closed the window. Such query windows are opened by clicking the following menu entries in the menus opened via the top buttons of the Coset Table Window: In the menu of `Settings`: `change default table size` (see 5.1.3), `extend table size` (see 5.1.4); in the menu of `File`: `read definitions from file` (see 5.1.10), `write definitions to file` (see 5.1.11), and `write standardized table to file` (see 5.1.12). Also such query windows are opened by the following "bottom buttons" of the Coset Table Window: `scroll by` (see 5.2.2), `scroll to` (see 5.2.1), `Felsch` (see 5.2.3), `fill gaps` (see 5.2.6), `fill rows` (see 5.2.5), `back to` (see 5.2.7), `HLT` (see 5.2.4), `short-cut` (see 5.2.9), and `mark cosets` (see 5.2.15).

All other windows

are ordinary **display windows** that can stay open while operations are caused by using other buttons or windows. Each of these display windows contains a top button `Sheet` that via a pull-down menu allows one either to write the contents of the display window to a PostScript file or to close this display window. See the paragraph **Button `Sheet`** in Section 5.1. A display window is opened by clicking the menu entry `show current settings` (see 5.1.8) in the menu opened via the top button `Settings`. Further display windows are opened by clicking the following bottom buttons in the Coset Table Window: `show rels` (see 5.2.10), `show subgrp` (see 5.2.11), `show defs` (see 5.2.12), `show gaps` (see 5.2.13), and `show coincs` (see 5.2.14). Further display windows can be opened by clicking with the right mouse button: coset numbers or dots in the Coset Table (see 4.1), entries in the Table of Definitions (see 4.8), entries in the List of Length 1 Gaps (see 4.9), and entries in the List of Pending Coincidences (see 4.10). Also the Subgroup Tables (see 4.7) and the Relation Tables (see 4.5) are display windows that can be opened by clicking the respective entries in the List of Subgroup Generators (see 4.6) or the List of Relators (see 4.4), respectively. The windows of Relation Tables and of Subgroup Tables differ from the other display windows in a technical way: Like the Coset Table Window, they stay open while the contents of the displayed tables changes. Moreover, the Relation Tables are scrolled parallel to the Coset Table (see 4.5).

Depending on the window system and window manager you use, placing a new window on your screen might be done automatically or might require you to use the mouse to choose a position for the window and pressing the left mouse button to place the window.

Making a window active is system and window manager dependent, in most cases you have either to move the pointer inside the window or you have to click the title bar of the window.

We recommend to switch on title bars for all windows in your window manager setup. This seems to avoid a small bug in at least one window manager we know of and makes it easier to move your windows around. For the `twm` window manager for example you can achieve this with the option `DecorateTransients`.

If you do not want to do this and experience problems with small dialog boxes that do not accept text entered via the keyboard, try to start up XGAP with the command line option `-W`. This invokes some code within XGAP which works around a bug in some window managers.

Please note that because of some 16 bit limitations in the underlying X toolkit the present implementation of XGAP may not be able to handle very large virtual windows properly. For instance, sheets containing more than 1630 coset definitions may be corrupt.

In the following chapters we will describe the possibilities to work with windows in a systematic way and by way of examples of the use of ITC.

1.5 Acknowledgements and Addresses

We thank Max Neunhöffer for providing the needed environment in his package XGAP and Thomas Breuer for frequent advice. A first version of ITC was implemented, still under GAP 3, by Ludger Hippe as part of his "Staatsexamensarbeit" in 1997.

Authors' e-mail addresses:

Volkmar.Felsch@math.rwth-aachen.de

Joachim.Neubueser@math.rwth-aachen.de

2 How to Start?

2.1 Starting XGAP

You begin an ITC session by starting XGAP. On most systems this is done by completing the provided line

```
user@host:~>
```

to

```
user@host:~> xgap
```

on the command line. Ask your system administrator if this does not work. This command will create a new window, the so called GAP Window, in which GAP is awaiting your input.

During a typical ITC session various windows will be opened to display the involved tables and further information. The size of the characters displayed in these windows and hence the sizes of the windows themselves are determined by the so-called **fonts** used.

The XGAP package knows five different fonts, `tiny`, `small`, `normal`, `large`, and `huge` (see the XGAP manual for details), but the ITC routines use only two of them, namely `small` and `normal`. So, if the default ITC window sizes are invonvenient for you, you need only to change the values of these two fonts. You may do this by starting XGAP with one or two additional arguments of the form "`-small` *font*" or "`-normal` *font*" where *font* has to be a font which is known to your system. You can get a list of all available fonts via the command

```
user@host:~> xlsfonts
```

Example:

```
user@host:~> xgap -small 6x12 -normal 8x13
```

2.2 Starting ITC

At the top of the GAP Window opened by XGAP you will see a "title bar" with three buttons: `GAP`, `Run`, and `Help`. Clicking either of these will pop up a menu, in which you may choose between several options.

You can always interrupt a computation by making the GAP Window active and pressing *CTRL-C*. You will get the same effect selecting `Interrupt` in the `Run` menu.

In the GAP Window, as usual, the GAP prompt

```
gap>
```

will indicate that GAP is awaiting your input.

If it is not loaded automatically, first load the ITC package by

```
gap> RequirePackage( "itc" );
```

See 74.3.1 in the GAP Reference Manual.

Then you may input a finitely presented group, say g, and a subgroup, say h, of it, the cosets of which you want to enumerate using ITC. The format of this input is the same as usual in GAP, see Chapter 45 of the GAP reference manual for the details.

ITC will in several of its windows have to display generators of the free group which is used to describe the presentation. Note that ITC will use the **names** of the generators of this free group, if these names consist of at most three symbols. Otherwise ITC will use names of the form $g1$, $g2$, ... up to at most $g99$ for the generators of the free group (if the group has more than 99 generators, an error message will be given). A choice of short and easily distinguishable names for the generators of the free group is hence advisable.

Now you are ready to start the ITC using the command `InteractiveTC`.

1 ▶ `InteractiveTC( `g`, `h` )`   F

Called with a group g and a subgroup h, this will open the "Coset Table Window", which really serves a multiple purpose:

– It shows the Coset Table and allows certain operations by mouse click in this Coset Table. We will describe the details in Section 4.1 of Chapter 4. In this chapter also a description of all other tables is given that can be shown on the screen by ITC.

– Directly below the Coset Table space is left for one row of text in which some warnings will be given (in red) by ITC if the user tries to invoke actions that in the present state are not possible, e. g. if the user tries to call the short-cut function while the tables have not yet closed (see Section 4.2).

– It provides up to date information on the state that an interactive coset enumeration has reached in the "Information Line" (see 4.3).

– It also provides a number of buttons which can be used in the interactive handling of a coset enumeration by ITC. We will describe this aspect in a separate chapter "Buttons and Menus" (Chapter 5).

The initial table size for the ITC Coset Table is set by default to 1000. You can change this default table size using the menu entry `change default table size` in the menu of the top button `Settings`, (see 5.1.3). If during a run of a CE the table size does not suffice, you can extend it using the menu entry `extend table size` (see 5.1.4) in the same menu.

3   The Basic Operations

For the meaning of terms used and the mathematical background we again refer to [Neu82]. In particular, we will quote as "Mendelsohn Condition" the condition formulated in Lemma 1 and Theorem 2 of that paper for eventual termination of a CE. In Sections 3.1 and 3.3 of this chapter, respectively, we describe briefly what the two basic operations "Making a Definition" and "Handling a Coincidence" do in ITC. In Section 3.2 we introduce some "Definition Strategies" from which a definition sequence can be put together if one does not want to define it step by step. Sections 3.4 and 3.5 describe methods by which one can modify existing definition sequences. The latter one may be of particular interest because it tries to "prune" a posteriori a definition sequence that has led to closure of all tables only after a number of coincidences had to be handled.

In the subsequent chapters we will then refer to these operations and strategies.

3.1 Making a Definition

ITC provides the possibility to define new coset numbers by clicking into certain places in various tables.

In each of these cases a new coset number will be defined to represent a coset that is the product (from the right) of an already defined coset with coset number i by an element x which is a generator or its inverse. If c is the highest coset number defined so far, this new coset number will be $c + 1$. It should be clear from understanding the idea of CE that this new coset number may in fact (by a different word in the generators) represent a coset for which a (lower) number has already been defined, and that this fact may only become apparent much later in the CE process.

The new coset number $c + 1$ is inserted into all relevant places in the coset table and all other tables. Also all consequences of this definition are entered into all tables, which in particular will get a new row $c + 1$. This newly defined coset number will occur in red colour in all places and so will the entries made as consequence of this definition. Entries that were previously coloured red get recoloured black.

Note however that as a consequence of this definition we may have obtained one or several coincidences. If this happens and if the coincidence handling is switched on (see `coincidence handling on`, 5.1.5, in the menu of the top button `Settings`) several coset numbers may get eliminated, and as the newly defined one may be among them, in fact it may not appear at all on the screen.

3.2 Some Definition Strategies

Already the first implementations of CE used different "strategies" for defining coset numbers, many interesting comparisons of their merits can be found in papers of George Havas and also in Sims' book [Sim94]. Not all but some are availavble in ITC, but those that are come in two forms.

- Either ITC can be asked to try to close the table following a chosen strategy. All commands to do this can be chosen from the menu pulled down using the top button `Close` (see 5.1.9 ff.).

- Or one can prescribe that ITC follows such a strategy only for a number of steps. This can in principle be done using bottom buttons. However since there are several choices of filling gaps of length 1, one

has to specify this choice before specifying the number of such "fill gaps" steps. This choice can be made using the possibilities offered in the menu pulled down by the top button `Settings` (see 5.1.7).

In the sequel we will describe the available strategies independently of their being used in one or the other way.

The **Felsch Strategy** (named after Hartmut Felsch, who first implemented it completely, see [Fel61], although according to John Leech, see [Lee84], the strategy was already used before in a partial implementation by Bandler, see [Ban56]) proceeds to define new coset numbers by row-wise from left to right filling the empty places of the Coset Table. It should be noted that of course quite different definition sequences can be obtained by just permuting the columns of the Coset Table. Using ITC this can either be achieved by renumbering the generators in the input of the group, or by filling lines in a different order "by hand". The Felsch strategy automatically fulfills the "Mendelsohn Condition" and hence is bound to stop eventually if the index of h in g is finite.

The **HLT Strategy** is named after B. Haselgrove, J. Leech, and H. F. Trotter, see [Lee84]. Its idea is to go for finding consequences soon by making definitions such that rows in Subgroup and Relation Tables get filled. Originally (when computers were still very slow) after a definition made in this way no search for consequences was started in order to save time, but only the consequences obtained "on purpose" by filling a row were noted. In ITC this is not the case, after each definition a complete search for all consequences is made, so that what we call HLT is not quite the original method. Note that by a pure HLT strategy the Mendelsohn Condition is not guaranteed. The CE may run into infinite loops although the index of h in g is finite. In programs such as ACE therefore in order to guarantee the Mendelsohn Condition the default HLT is modified by closing a row of the Coset Table after all corresponding rows of Relation Tables have been closed. Note that this is not done in ITC because we want to provide the user with the possibility to experiment with clearly understood alternatives, not with a safe program to enumerate millions of cosets.

Filling gaps of length 1 in relation (or subgroup) tables has in particular been propagated by George Havas. The idea is that this is the "cheapest" way to obtain consequences. As will be seen in the examples, in particular that of the Fibonacci Group $F(2,7)$, the result heavily depends on the choice of one of the sometimes many gaps of length 1 to be filled. (Note in particular that these can occur not only in the Relation Tables that are shown by ITC, but also in relation tables belonging to cyclicly permuted relators.) To explain the different strategies available in ITC for filling gaps of length 1 we have to introduce the notions of the equivalence class and of the weight of a gap of length 1.

Let a gap of length 1 in a Relation Table or Subgroup Table between coset numbers i and j be filled by defining a new coset number k in this place. Then, still excluding the newly introduced kth row, this fills two places in the Coset Table which we call a "pair" of "dots" (These places will indeed be marked by dots in the ITC Coset Table). On the set of all dots in the Coset Table we define the finest equivalence relation in which "paired" dots are equivalent. So each gap of length 1 in a Subgroup or Relation Table determines a unique equivalence class of dots, and the length of this class is defined to be the "weight" of the gap of length 1. For each class we define its "representative" to be the first of its dots with respect to the usual row-wise from left to right ordering of positions in the Coset Table. (Instead of speaking of equivalnce classes of dots, we will sometimes also speak of equivalence classes of gaps of length 1.)

For choosing the next gap of length 1 we can now define several strategies:

strategy 1 (first gap)
> Here the gap belonging to the first dot in the row-wise from left to right scan of the Coset Table (irrespective of its weight) is filled.

strategy 2 (first rep of maximal weight)
> Here among the equivalence classes of gaps of maximal weight the one is chosen for which the representative dot occurs first in the row-wise from left to right scan of the Coset Table.

strategy 3 (last rep of maximal weight)
> Here among the equivalence classes of gaps of maximal weight the one is chosen for which the representative dot occurs last in the row-wise from left to right scan of the Coset Table.

3.3 Handling a Coincidence

Whenever a row in a Subgroup or Relation Table closes because of a new definition made we will get a consequence, saying that the product of a coset with number i by x, a generator or its inverse, equals a coset with number j. This fact may not yet be "known" in the Coset Table, in which case it is entered into the Coset Table, or it may already be known in exactly this form in the Coset Table, in which case no action is needed. However the Coset Table may also contain already the information that the product of the coset with number i by x equals a coset with number k different from j. In this case we learn that j and k really denote the same coset.

When such a "coincidence" is detected, and if k is bigger than j, say, all consequences of the fact that then the jth and the kth row of the Coset Table must describe the same facts are saved in the j th row and other rows of the Coset Table, before the kth row is declared "dead". (See [Neu82] for a simplified description of that process and a justification why it preserves all essential features of a coset table.)

This is done automatically in ITC if the coincidence handling is switched on, it can be done "by hand" if it is switched off (see 5.1.5).

Note that as a consequence of handling a particular coincidence new ones may be detected. All these are worked off automatically or must be worked off "by hand", respectively, before new definitions can be made.

3.4 Sorting Definitions

The purpose of this method is to modify definition sequences in order to be able to compare different such sequences. This will be done by constructing from a given definition sequence a "canonical" one by reordering and renumbering coset numbers. The method proceeds as follows.

In a first **step** all cosets obtained from the coset with number 1 by multiplication with generators or their inverses will be arranged in the order in which they occur in a row-wise scan of the Coset Table from left to right. Then the first of these is taken and an analogous step is performed with its images under all the generators and their inverses. The coset numbers obtained in this way are appended to the sequence obtained in the first step. After that the same is done again and again in obvious analogy to this second step until all coset numbers in the Coset Table have been rearranged by this procedure.

Then the coset numbers are changed according to this new ordering.

The definition sequence thus obtained is now processed by CE (with prescribed definition sequence). In this CE some of the coset numbers may get eliminated by coincidences, so that the resulting definition sequence may be shorter (but can never get longer) than the original one. Moreover because of such a coincidence the intended ordering may again be disturbed. If that is the case, the whole **procedure** – as described up to this point – is repeated. However since the second definition sequence will be lexicographically earlier than the first one we must arrive at a stable sequence after a finite number of repetitions of such procedures.

This "Sorting Definitions" is applied to the definition sequence after all tables have been closed. Note that if the original definition sequence will have been obtained by a pure Felsch strategy, it will not be changed by this Sorting Definition method.

We remark that this method can be used independently of the Short-cut method (to be described next) before or after it. Combining the two methods may in fact improve but also deteriorate results.

3.5 The Short-cut Method

It is easy to construct examples of presentations of finite groups such that any coset enumeration process will have to define some "redundant" coset numbers, that is, will have to handle some "coincidences" before finding the order of that group. An easy example is the trivial group presented on one generator x by the relators x^5 and x^7. In such cases it is a natural challenge to find definition sequences that reach the result with few coincidences showing up. No method is known that will construct a definition sequence with a number of coincidences guaranteed to be minimal. The package ACE by George Havas and Colin Ramsey (see [Ram99]) contains various strategies that have proven to be very efficient in constructing from scratch definition sequences involving few coincidences.

Here we describe a different approach, namely to try a posteriori to construct a definition sequence with few coincidences by "pruning" an already existing definition sequence that had led to coincidences. We call this the "Short-cut method".

Let a definition sequence be given that at the end after defining a last coset number n leads to closure of the tables. We trace back the sequence of those definitions from n to coset number 1 that were actually used to reach coset number n. Say k coset numbers occur in that sequence which we number $i_1 = 1, \cdots, i_k = n$. We mark these k coset numbers as being "indispensable" and move their sequence to the beginning of a new definition sequence. In general not all coset numbers defined in the CE will occur in the sequence of indispensables, so we complement the sequence of indispensables by the remaining coset numbers in the order in which they occurred in the original definition sequence. If we now start a CE with that new definition sequence, it must definitely close after having worked through this whole series, however it may in fact lead to closure after only a subset of these definitions.

With the definition sequence obtained from this second CE (that at the beginning will still retain indispensables) we now repeat the same procedure starting again from the last coset defined before closure, if this was not yet marked indispensable. The subsequence thus obtained is again marked indispensable and moved to the beginning of the previously existing definition sequence. Behind these, we put again all remaining coset numbers of the previously obtained sequence and then start a CE again. (Note that those marked "indispensable" in the first "loop" now come behind those marked "indispensable" in the second "loop" and that some of the "indispensables" of the first "loop" may now in fact vanish in this second CE).

Obviously this process can be iterated as long as not all coset numbers in one of the sequences obtained on the way are marked indispensable. Once this is the case, the Short-cut method stops.

ITC also allows one to prescribe the number of "loops" to be performed.

As can be seen from the examples, the Short-cut method works with surprising success. In the case of some presentations that for a long time have been used as challenges for finding the shortest definition sequence leading to closure, the Short-cut method even improved on results that had been found "by hand" with quite some effort by experts on CE.

However it should be emphasized again that no claim is made that the method will find the minimal number of cosets needed to reach closure. In fact the examples given show that the result of the Short-cut method depends on the definition sequence that it starts from.

Although yielding its result only a posteriori, we hope that beyond answering challenges, finding a short definition sequence may become of use in algorithms such as the modified TC if this is implemented to follow a prescribed definition sequence.

4 The Tables

Usually table entries in all tables are black. Red indicates that these entries were the result of the last step that has been performed by ITC. Coset numbers can also be coloured green using the `mark cosets` bottom button (see 5.2.15) or the `show defs` bottom button (see 5.2.12). In case of conflict green has preference to red.

As a general rule we state that clicking some place in one of the **tables** with the **left** mouse button will result in an action in the process of CE, e. g. in making a definition.

There are only very few cases where clicking with the **right** mouse button in a table will have an effect:

– with clicking a coset number or a dot in the Coset Table (see Section 4.1),

– with clicking on an entry of the Table of Definitions (see Section 4.8),

– with clicking on an entry of the List of Length 1 Gaps (see Section 4.9 and 5.2.13), and

– with clicking on an entry of the List of Pending Coincidences (see Section 4.10).

In each of these cases clicking with the right mouse button will result in ITC displaying further information in a new display window.

4.1 The Coset Table

The main table in any coset enumeration is the coset table. Its rows are numbered by natural numbers, the "coset numbers", which are defined in the process of the coset enumeration. Remember that coset numbers denote cosets defined by representatives given by words in the generators, but until a coset enumeration is finished, different coset numbers may represent the same coset (by different words).

Note that ITC displays on the screen only rows for "still alive" coset numbers, that is coset numbers that have not been eliminated by coincidences.

So the first column of the ITC Coset Table displays the list of presently "still alive" coset numbers. The further columns of the ITC Coset Table are indexed by the generators of the finitely presented group g and their inverses.

Entries in an ITC Coset Table can either be empty, or a coset number, or a dot. An entry j in row i and column k means that the coset with number i multiplied by the generator or inverse generator heading column k yields the coset with number j. An empty entry indicates that nothing is yet known about the respective product. A dot as entry indicates that making a definition in this place will close at least one row in a subgroup or a relation table, i.e. it belongs to a "gap of length 1" (see Section 3.2).

Let h be the subgroup of the group g whose cosets are to be enumerated. At the beginning of an enumeration ITC fills automatically those entries of the Coset Table (as well as of Subgroup and Relation Tables) that are consequences of coset number 1 always being given to the subgroup h.

Clicking with the **left** mouse button an empty place or a place with a dot in the Coset Table will make a definition in this place. That is: Let c be the last coset number defined, then a click in the place in row i and column headed by generator or inverse generator x will define a coset number $c + 1$ that represents the

product of the coset with number i by x. See Section 3.1 for details of the changes (marked in red) in the tables caused by making such a definition.

At the same time new dots may occur in the coset table, representing places newly found to allow filling gaps of length 1 in one of the Relation Tables or Subgroup Tables. These new dots will also be coloured red.

Clicking with the **right** mouse button a dot will open a window which will show a list of all those definitions that are equivalent to the one belonging to this place in the sense that they will close the same length 1 gaps in Relation and Subgroup Tables. (Compare also 3.2 and 5.2.13).

Clicking with the **left** mouse button the coset number at the beginning of a row of the Coset Table will result in closing this row by new definitions. (This way you can e. g. easily fulfill the Mendelsohn Condition for this row requesting that images of that coset under multiplication by all generators and their inverses must eventually get defined).

Clicking with the **right** mouse button the coset number at the beginning of a row of the Coset Table or any already defined coset number in the Coset Table will open a window in which a coset representative of the respective coset will be displayed as a word in the generators of g and their inverses. Clicking the same coset number once more (in the same or in another position of the Coset Table) will close the coset representative window again. (In Section 4.8 we will see that the same coset representative window can be switched on or off also by clicking the respective row in the Table of Definitions.)

Note that only up to 30 rows of the Coset Table are displayed simultaneously on the screen but that the Coset Table can be scrolled using the bottom buttons `scroll to` (see 5.2.1) and `scroll by` (see 5.2.2).

4.2 Warnings

Directly below the Coset Table a row is left open in which warnings will flash up in red if the user is trying to start an action that is impossible or does not make sense in the present situation, so that e. g. certain buttons are disabled. These are:

`There are no subgroup generators`
> if the user clicks the bottom button `show subgrp`, but there are no subgroup generators specified, i. e. one is performing a CE of the cosets of the trivial group.

`There are no gaps of length 1`
> if the user tries to show or to fill gaps of length 1 using the bottom buttons `show gaps` or `fill gaps`, respectively, but no such gaps of length 1 exist in any of the Subgroup or Relation Tables.

`There are no more gaps of length 1`
> if the user had requested to perform a prescribed number of steps filling gaps of length 1, but the supply of such gaps ran out before this number of steps had been performed.

`There are no pending coincidences`
> if the user tries to see the list of coincidences while no such are known.

`There are pending coincidences`
> if the user tries to make any definition while coincidences are still pending, which may be the case only if the automatic coincidence handling is switched off. (Note that the user is informed anyhow of the fact that coincidences are pending in the "Information Line").

`The tables are closed`
> if the user tries to make any definition after the tables are closed already.

`The tables are not closed`
> if the user tries to invoke the Short-cut method before the tables are closed.

`Insufficient table size`
> if the coset enumeration comes to a halt without closing the tables because the size for the Coset Table was too small. In this case this size has to be extended using the menu entry `extend table size` in the menu popped up via the top button `Settings` (see 5.1.4).

`Illegal argument`
> if an illegal answer has been typed into a query window, e. g. if the number of steps requested in the query window opened by clicking the bottom button `Felsch` is given as a negative integer.

`Illegal coset number`

`This command has no effect`

The warning message will vanish as soon as the user performs the next allowed action.

4.3 The Information Line

Underneath the Coset Table you see a line which will all the time display three numbers that characterize the present state of the coset enumeration:

`Defined:` i
> means that so far i coset numbers have been defined altogether (some of which may already have been eliminated again by the coincidence routine).

`Deleted:` j
> means that so far j of these coset numbers have been eliminated again through the handling of coincidences (see also Section 3.3 and 5.2.14).

`Alive:` k
> means that k coset numbers have survived the elimination process of the coincidence handling (if coincidences did occur at all). So obviously $k = i - j$. Note that during a run of ITC there is no change (renumbering) of coset numbers after a coincidence has been handled, i. e. the sequence of coset numbers will always be an ordered sequence of natural numbers starting with 1, but if coincidences have been handled, it will have "holes", i. e. no longer be consecutive. While in fact in the internal data structure for the coset table these "dead" cosets still exist, in the Coset Table (and Relation Tables) occurring on the screen these lines will be suppressed, so that at any given moment the actual number of lines in the Coset Table and the Relation Tables that can be seen on the screen will be k, (not i). This has some impact on the scrolling of the Coset Table (compare 5.2.1 and 5.2.2).

Behind these three numbers the Information Line will at certain instances also display warnings:

– When the CE has come to an end, that is, when all tables have been closed with no coincidences pending, the text `Tables Closed` is shown in red.

– If the handling of coincidences has been switched off using `coincidence handling off` (see 5.1.5) in the menu of the top button `Settings` the text `Coincidence Handling OFF` is shown in red until a coincidence is really encountered (if any occur at all). Then the process will come to a halt and the text shown will change for the warning `Pending Coincidences`, again shown in red. Note that this in particular means that the tables will not have closed and the CE has not been finished yet. If the CE process is resumed after the pending coincidences have been worked off, also the text `Coincidence Handling OFF` will reappear as long as coincidence handling remains switched off and not all tables have closed.

– If the definition sequence has been changed using the Short-cut method (see 5.2.9 as well as Section 3.5) this will be indicated by the text `Short-Cut` shown in red, followed by the number of "loops" that the Short-cut method has performed (see Section 3.5). Analogously `Tables sorted` is shown after `sort defs` (see 5.2.8) has been applied.

4.4 The List of Relators

The List of Relators is shown in a display window that pops up after clicking the (white) bottom button `show rels` (see 5.2.10).

Clicking one of the relators opens a window for the respective Relation Table (see 4.5).

4.5 The Relation Tables

For a relator that is the product of n generators or inverse generators the Relation Table is a table with $n + 1$ columns. The vertical lines between the columns of the table are headed by the factors of the relator. Each row has in its first and last place the same coset number. Further entries can either be empty or be occupied by a coset number.

Assume that at a certain time coset numbers 1 to c have already been defined and consider an empty entry in a Relation Table that is either the first empty one or the last empty one in its row. Then clicking such an empty space defines coset number $c + 1$ in this place.

To be more explicit:

Let i be the coset number standing before an empty place and x be the generator or inverse generator heading the vertical line in front of the empty place then by clicking this empty place $c + 1$ is defined to be the coset number for the coset that is obtained as the product of the coset with coset number i by x.

Analogously let i be the coset number standing behind an empty place and x be the generator or inverse generator heading the vertical line behind the empty place then by clicking this empty place $c + 1$ is defined as the coset number of the coset obtained as the product of the coset with coset number i by x^{-1}.

See Section 3.1 for more details on the resulting changes of the tables.

Clicking the **first** or the **last** entry of a row (that are always filled by the same coset number as soon as this row exists) will result in **filling this whole row** in an analogous manner starting from the left. The last coset number defined in this process (i.e. the rightmost one) and the consequences of its definition get marked in red.

For the possibility to fill **all** ith rows in all Relation Tables you can use the green bottom button `fill rows` (see 5.2.5).

Clicking on some other place already filled by a coset number will have no effect. Also clicking on an empty place that is neither the first nor the last empty one in its row will have no effect.

Unlike most of the other display windows, but in line with the Coset Table Window, the windows of the Relation Tables do not vanish when the contents of the Relation Tables changes but the tables change in their windows. Note that (as with the Coset Table) only up to 30 rows of the Relation Tables are displayed simultaneously on the screen but that the Relation Tables can be scrolled parallel to the Coset Table using the bottom buttons `scroll to` (see 5.2.1) and `scroll by` (see 5.2.2).

4.6 The List of Subgroup Generators

The list of generators of the subgroup h is shown in a window that springs up after clicking the (white) bottom button `show subgrp` (see 5.2.11).

Clicking one of the subgroup generators opens a window for the respective Subgroup Table (see 4.7).

4.7 The Subgroup Tables

Except that they have only one row that begins and ends with the coset number 1, Subgroup Tables are exactly like Relation Tables. All that has been said about making definitions in Relation Tables holds for Subgroup Tables as well. Also unlike most of the other display windows, but in line with the Coset Table Window, the windows of the Subgroup Tables do not vanish when the contents of the Subgroup Tables changes but the tables change in their windows.

4.8 The Table of Definitions

Whenever the `ITC` performs a coset enumeration it saves the occurring sequence of definitions of coset numbers in a **List of Definitions**. This list is not affected by coincidence handling, however it is changed if you make new definitions or retrace to a previous state of the definition sequence using the `back to` bottom button (see 5.2.7) or call the Short-cut algorithm (see 5.2.9) or the "Sorting Definitions" (see 5.2.8). To some extent the List of Definitions reflects the history of the enumeration. It will probably be mainly of interest for use in further computations and can for that purpose be written to a file (see `write definitions to file`, 5.1.11, and `read definitions from file`, 5.1.10, of the top menu popped down by clicking the top button `File`).

The current state of the enumeration is reflected by a second list which we will call the **Table of Definitions**. It contains only the definitions of those coset numbers which are "still alive", i. e. which have not vanished by coincidences. and all coset numbers involved in these definitions which are not alive any more are replaced by their alive representatives. So the Table of Definitions also changes whenever a coincidence is handled.

Upon clicking with the **left** mouse button the (white) bottom button `show defs` (5.2.12) a window appears which displays the Table of Definitions or, to be more precise, an extract of that table consisting of the up to 30 lines corresponding to the cosets currently shown in the Coset Table Window. Like the Relation Tables (see Section 4.5) this window does not vanish when the contents of the Table of Definitions changes but the table changes in the window, and whenwever the Coset Table is scrolled using the bottom buttons `scroll to` (see 5.2.1) and `scroll by` (see 5.2.2) it will be scrolled in parallel.

Clicking with the **left** or with the **right** mouse button a definition **in the Table of Definitions** will open a window in which a coset representative of the coset defined here will be displayed as a word in the generators of g and their inverses. The coset representative displayed is the one that is obtained by retracing in the List (not the Table) of Definitions from this coset number to the coset number 1. A second click on the same definition will close the coset representative window again. Note that the coset representative windows can also be opened or closed by clicking a coset number in the Coset Table with the right mouse button (see Section 4.1).

4.9 The List of Length 1 Gaps

For some background definitions for "gaps of length 1" see Section 3.2.

On clicking the (white) bottom button `show gaps` (see 5.2.13), either, if no gaps of length 1 exist, only a warning `There are no gaps of length 1` appears in red below the Coset Table, or a window springs up that lists representatives of the equivalence classes of gaps of length 1 in Subgroup and Relation Tables.

A row in this list has the form i: w [l, x] or i: w [l, x^{-1}] respectively. Here i is just the number of the list entry in a consecutive numbering and w is the weight of the respective equivalence class of gaps of length 1. The remaining information describes the representative of that equivalence class (it indicates that defining a new coset as the product of the coset with number l and generator x, or x^{-1}, respectively, will close gaps of length 1 of weight w in Subgroup and Relation Tables).

The idea of giving this information is that the "weight" w may be a guide for the choice of definitions that have a strong influence on the progress of a CE.

By clicking on such a row with the **left** mouse button this definition will be made and together with its consequences inserted into all relevant places in the tables. See Section 3.1 for more details.

Clicking on one of the entries in this list with the **right** mouse button will open a further window in which all equivalent definitions are listed in the form [l, x] or [l, x^{-1}], respectively.

Clicking on any of these, again with the **left** mouse button, will result in making this definition. See again Section 3.1 for details.

4.10 The List of Pending Coincidences

Clicking the (white) bottom button `show coincs` (5.2.14) with the **left** mouse button you either get the warning `There are no pending coincidences` shown in red below the Coset Table, or a window springs up in which the list of all pending coincidences is shown. These appear in the form $i = j$ meaning that the coset numbers i and j really denote the same coset. If i is the larger integer then clicking such a line in the List of Pending Coincidences will eliminate i, that is replace it by j in all occurrences in all tables and then suppress the ith lines in all tables. See also Section 3.3.

As a consequence, more coincidences may be found, so that the List of Pending Coincidences may have become longer.

In case coincidence handling is switched off (see `coincidence handling off`, 5.1.5 in the menu of the top button `Settings`), the CE process will stop automatically as soon as a coincidence is encountered and then the window for the List of Pending Coincidences will spring up automatically.

Since a coincidence really means that ITC has found two representatives, say r and s, of the same coset, given as words in the generators of the group, the quotient $r * s^{-1}$ will be an element of the subgroup h. Clicking this coincidence with the **right** mouse button will open a window in which the subgroup element $r * s^{-1}$ will be displayed as word in the generators of g and their inverses.

As in particular George Havas has pointed out in several papers (see for instance [HR99]) it can be helpful in some "hard" coset enumerations to add subgroup words obtained in this way from coincidences to the subgroup generators in the GAP input of the given subgroup h and to restart ITC.

5 Buttons and Menus

The Coset Table Window will initially show you a coset table with columns for each of the generators of the presently investigated group g and their inverses, in which only the consequences of giving the subgroup h the coset number 1 are inserted (so that at the beginning it has only one row). We have already described in Chapter 4 how to work interactively by mouse clicks in this table.

Also in that chapter (see Sections 4.2 and 4.3) we have described the warnings that may appear directly below the Coset Table and the "Information Line" that is placed further below the coset table.

However the Coset Table Window also provides via buttons and menus access to much of the functionality offered by ITC.

The Coset Table Window has a row of four buttons at the top: `Sheet`, `Settings`, `Close`, and `File`. We will refer to these as the **top buttons**. Clicking one of these will pop up a menu from which you can select actions in the course of a coset enumeration. You do this by holding the left mouse button while moving the pointer to the respective entry in the menu and releasing the mouse button there. Then the respective action of the program will be executed while the menu vanishes.

This action may consist in popping up yet another (query) window into which you have to write further (e. g. numerical) instructions or in setting modes for the CE or in immediately performing certain changes in the coset table and other tables.

This way of selecting actions may be felt to be uncomfortable and hence for convenience the most frequently used actions occurring during a coset enumeration can be prompted more directly by clicking one of 18 buttons provided at the bottom of the Coset Table Window. We will refer to these as the **bottom buttons**.

In the section "The Top Buttons" (5.1) we will describe the menus belonging to the buttons at the top of the coset table.

In a further section "The Bottom Buttons" (5.2) we will describe the use of the bottom buttons.

Many of the windows to be described will contain two further buttons `ok` and `cancel`. Clicking `ok` means that you want the order executed that is described in the window while `cancel` means that you do not want this execution. In any case, having pressed one of this buttons, you will afterwards return to where you called this window from.

Note that **buttons** are in general clicked with the **left** mouse button.

There is one exception where clicking a bottom button with the **right** mouse button will have an effect namely with showing the definitions made. This is explained in the description of the effect of the bottom button `show defs` (see 5.2.12).

5.1 The Top Buttons

Button `Sheet`

Note: **This button also appears as a top button in each of the "display windows".**

Display windows (see Section 1.4) are opened by clicking the following bottom buttons of the Coset Table Window: `show rels` (see 5.2.10), `show subgrp` (see 5.2.11), `show defs` (see 5.2.12), `show gaps` (see 5.2.13), and `show coincs` (see 5.2.14), as well as in the display window opened by clicking the menu entry `show current settings` in the menu popped up via the top button `Settings` (see 5.1.8).

The menu popped up by the button `Sheet` has only two entries:

1 ▶ `save as postscript`

> This allows you to save the contents of the respective window (e. g. the Coset Table Window) as a PostScript file. Releasing the left mouse button on this entry will pop up a further menu which allows you to enter the name of the file to which you want this to be written. You are assisted doing this by a list of names of files that exist in your directories, and from which you can choose. The menu also offers you three more buttons: `ok`, `cancel`, and `home`. Choosing `home` will insert the name of your home directory into the provided field while `ok` starts the action after you have finished the name of the target file, and `cancel` brings you back to the window in which you clicked the button `Sheet`.

2 ▶ `close graphic sheet`

> This closes the display window and brings you back to the window from which you had opened it.

Button `Settings`

The menu popped up by this button has ten entries. Two of these allow one to influence the size of the table provided for the coset enumeration. Seven further ones allow one to turn on or off certain modes to be followed during the enumeration. These modes can be turned on or off at the beginning, but can also be changed later on during the enumeration. The last menu entry allows one to open a window in which the presently set parameters and modes are shown.

3 ▶ `change default table size`

> This opens a query window (see Section 1.4) in which you can change the default table size from the original value of 1000 to any positive integer you want by typing the desired size into a field provided for this purpose, but consider that unnecessarily high values may cause garbage collections. Clicking `ok` means that you accept the value now given in the field and return to the Coset Table Window, clicking `cancel` brings you back there without having changed the existing default table size. You can change your decision by going to that window again. Note that changing the default value while a CE is under the way does not change the present table size. If you want to or have to change the table size during a coset enumeration, you can use the next menu entry `extend table size` (see 5.1.4).

4 ▶ `extend table size`

> The text in the query window (see Section 1.4) tells you the present table size and proposes twice that size as the value to which it should be changed. You can again alter that value. `ok` and `cancel` have the same meaning as before.

Note that an analogous window for extending the table size will spring up if in the coset enumeration you have exhausted the preset table size. If you extend the table size and leave by clicking `ok` the CE will immediately be resumed with the extended table size being provided. If you do not extend the table size in such a case, ITC will refuse to accept or make further definitions. If you then extend the table size using `extend table size` as described in 5.1.4 the calculation can be resumed but must be restarted again.

5 ▶ `coincidence handling off`
   ▶ `coincidence handling on`

> ITC allows one to follow stepwise the handling of coincidences if such occur. Since however this can be a fairly slow procedure that one might not want to follow in detail, this feature can be turned on and off by these two menu entries. If coincidence handling is switched off any process will automatically come to a halt as soon as a coincidence is encountered. At that time a window with the list of pending coincidences will spring up and can then be used to eliminate coincidences one by one in any wanted sequence "by hand" (see 4.10). Also in the coset table (in addition to the last definition made) all rows get marked in red that are up for elimination. Note that while coincidences are pending the same coset number can occur more than once in a column of an ITC Coset Table. By default coincidence handling will be switched on if you start with a new group from the GAP window, however if you make a new attempt with the same group having finished the last one by the `clear` button (see 5.2.16) you will retain the previous choice.

6 ▶ `echo on`
   ▶ `echo off`

> When you use a mouse click to perform any of the operations of ITC this will be documented by a line of the form `gap> PointerButtonDown(`$x,y,\ldots$`)` or `gap> MenuSelected(`$u,v,\ldots$`)` in the GAP window, where e. g. the integers x, y, ... specify the position in the window in which you have clicked. This, however, does not help you easily to understand what you have done. It is not even helpful for re-doing what you have done, because in a repeat the windows may sit in different places. On the other hand in a long interactive run you may want to be able to reconstruct your actions, be it – horribile dictu – that you have run into a bug that you want to report, or - rather better - that by a stroke of luck you found the result in a particularly nice and short way. For this purpose you can click the menu entry `echo on`. Having done this the GAP window will after each of these rather ununderstandable messages print with the prefix `>>` readable information on the button or menu entry, respectively, that you had clicked, in many cases simply the text written in that button or menu entry. Since you can write in parallel everything occurring in the GAP window to a file using the `LogTo` command (see 9.7.5 in the GAP reference manual) you can thus keep a record of your interactive actions. If you do not want this feature, you can switch it off again.

7 ▶ `gaps strategy 1 (first gap)`
   ▶ `gaps strategy 2 (first rep of max weight)`
   ▶ `gaps strategy 3 (last rep of max weight)`

> These strategies have been explained in Section 3.2. You have the option to **preset the strategy** of selecting the next gap of length 1 using these menu entries. Then they can be executed using the bottom button `fill gaps` (see 5.2.6). In the query window opened by clicking this bottom button you can prescribe **how many** "fill gaps" steps you want to make following the preset strategy. By default strategy 1 will be chosen.

8 ▶ `show current settings`

> Selecting this menu entry opens a display window in which the chosen default table size and the current table size as well as the choices of modes (`coincidence handling on/off` and `echo on/off`) and of the strategy for choosing the next gap of length 1 are shown. As long as this window remains open it will be updated whenever any of these values is changed.

Button `Close`

The menu popped up by this button allows one to choose various strategies for trying to finish the coset enumeration. The present menu entries are

9 ▶ `close table by Felsch`
  ▶ `use gaps strategy 1 (first gap)`
  ▶ `use gaps strategy 2 (first rep of max weight)`
  ▶ `use gaps strategy 3 (last rep of max weight)`
  ▶ `close table by HLT`

> All these have been described in principle in Section 3.2. Note however that the "closing gaps of length 1" strategies automatically use Felsch steps if and as long as no gap of length 1 exists at a certain stage. If you want to see exactly how this happens you have to use the `fill gaps` command (see 5.2.6) with some high number of wanted steps and to help it with Felsch steps thrown in interactively if it gets stuck because of lack of gaps of length 1. We will demonstrate this in the first of the examples. Note that all but the first strategy do not guarantee the Mendelsohn Condition.

Button `File`

The menu popped up by this button has three entries which concern reading from and writing to files important information that may have been obtained by a longer interactive coset enumeration:

10 ▶ `read definitions from file`

> Releasing the mouse button at this entry opens a query window (see Section 1.4) in which you can write the name of the file. You are assisted in doing that by a list of names of files that exist in your directories, from which you can choose. Note that this choice is disabled (and the menu text `read definitions from file` coloured grey instead of black) as soon as you have started a coset enumeration so that reading in another definiton sequence does not make sense any longer.

11 ▶ `write definitions to file`

> If you have constructed a definition sequence that you want to keep in order to possibly reuse it, you can write this definition sequence to a file. Releasing the mouse button at this entry opens a query window (see Section 1.4) in which you can write the name of the file. Again you are assisted by being offered some filenames from which you can choose. Note that a definition sequence can be written to a file at any time during a CE, so that you can use it to proceed in different ways from a certain state that you have reached. However note that this menu entry is disabled (and the menu text `write definition to file` coloured grey instead of black) as long as no definition sequence exists yet.

12 ▶ `write standardized table to file`

> As mentioned e. g. also in Section 4.3 there is no renumbering of coset numbers in case some of them are eliminated by the handling of coincidences. That means that at the end of a coset enumeration that did involve handling of coincidences the coset numbers will form a sequence of natural numbers that starts with 1 but has "holes". This is inconvenient, in particular if one wants to consider the columns of the coset table as the permutations of the cosets under right multiplication by the respective generator (or inverse generator). Therefore after a CE has finished it is possible to write a standardized coset table (in which in particular the coset numbers now run from 1 to the index of h in g) to a file, which then can be processed in any wanted way by GAP. Releasing the mouse button at this entry will open a query window (see Section 1.4) in which you can write the name of this file. Again you are assisted by being offered file names from which you may choose. Note that this menu entry is disabled and the text `write standardized table to file` coloured grey instead of black as long as the tables have not closed. That is, only closed coset tables can be standardized by ITC. A coset table that has been written this way to a file with name *filename* can be read into GAP and assigned to a variable *costab* via

> *costab*`:= ReadAsFunction(` *filename* `)();`

5.2 The Bottom Buttons

The different kind of functionality of these 18 buttons is to some extent indicated by their colours.

The first two (blue) buttons just allow one to scroll to the lines of the coset table that one wants to work with, if it contains more than 30 "alive" rows.

1 ▶ `scroll to`

> Upon clicking this button a query window (see Section 1.4) will be opened in which you can specify **by its number** the row in the Coset Table that you want to see. The Coset Table will be scrolled up or down in such a way that the wanted row appears in the middle of the displayed part of the Coset Table. As a default, the number of the last still alive row of the Coset Table will be offered to you. If the specified coset number is no longer "alive" the maximal alive coset number below the specified one is taken instead. Any displayed Relation Tables or the Table of Definitions will be scrolled in parallel to the Coset Table.

2 ▶ `scroll by`

> Upon clicking this button a query window (see Section 1.4) will be opened in which you can enter into a field provided for the purpose the number of rows by which you want to scroll up or down in the coset table. Positive integers mean scrolling up, that is to higher coset numbers, negative integers mean scrolling down, that is to lower coset numbers. Originally the value 20 (that you may change) is offered in this field. If you click this button a second time, the number that you had accepted last time will be offered again, thus allowing you to glance through the Coset Table at certain intervals without having to type in a new number each time, Note that here **actual rows shown in the tables** are counted. That is, if some rows have been suppressed by coincidence handling they are not counted, so that you may end up with rows of higher numbers than you expected. Any displayed Relation Tables or the Table of Definitions will be scrolled in parallel to the Coset Table.

The next seven (green) buttons all give access to certain strategies to be followed through several steps in a coset enumeration. See Section 3.2 for details.

3 ▶ `Felsch`

> The "Felsch" strategy for performing a coset enumeration makes a sequence of definitions by row-wise filling empty places in the coset table and by searching for all consequences after each definition which are then immediately worked in as well as all coincidences that may be found. Upon clicking this button a query window (see Section 1.4) will be opened, in which you can choose how many definitions you want to have made following this strategy. As default (that you can change) the number 1 is offered; choosing this is hence tantamount to making a definition in the first empty place in the coset table in a row-wise scan.

4 ▶ `HLT`

> The "HLT" strategy proceeds by filling rows of Subgroup and Relation Tables (see Section 3.2 and note in particular that as explained there the HLT strategy of ITC deviates in some respect from what is usually meant by the term). Upon clicking this button a query window (see Section 1.4) is opened in which you can specify how many definitions you want to make using this strategy. If you want to **close** the tables using a HLT strategy, you can use the menu entry `close table by HLT` (5.1.9) in the menu opened by the top button `Close`.

5 ▶ `fill rows`

> This button allows you to fill at once in all Relation and Subgroup Tables all rows that belong to one or more prescribed coset numbers. Upon clicking this button a query window (see Section 1.4) will be opened in which you can specify the wanted coset numbers, separated by commas. As a

default the smallest number i is offered for which not all ith rows of Relation and Subgroup Tables are filled yet. Note that you can only fill rows that do exist, that is rows of a coset number that has already been defined and has not been eliminated by coincidence handling. If you violate this condition, your order will be ignored. If you want to fill a row only in a particular Subgroup or Relation Table or in the Coset Table you can do this by clicking at the beginning of the respective row (see 4.1 and 4.5). Specifying the (non-existing) coset number 0 allows you to fill just the (first and only) rows of the Subgroup Tables without filling rows in any of the Relation Tables.

6 ▶ `fill gaps`

Closing rows in relation or subgroup tables in which only one place is open, is the easiest way to get immediate consequences. Such empty places are called **gaps of length 1**. Often several choices of such "gaps of length 1" to be closed are possible and ITC offers various strategies to make this choice (see Section 3.2). (Remember – compare Section 3.2 – : by inspecting the Relation Tables that you can display by ITC you may not see all these gaps of length 1, since ITC does not display all the relation tables that can be obtained by cyclic permutation of the relators in the given presentation.) These strategies can be set via the top button `Settings` (see 5.1.7); by default strategy 1 will have been set. Clicking the button `fill gaps` will open a query window (see Section 1.4), in which you can specify how many times you want to close a gap of length 1 that is chosen by the selected strategy. By clicking the button `show gaps` (see 5.2.13) you can learn which gaps of length 1 are open at a given time. However note that by closing a particular gap of length 1 often new gaps of length 1 will be created, so that by entering a number, say ng, bigger than the number of presently existing gaps of length 1 into this window you will start off what might be called a "pure minimal gap strategy". This is often a very good strategy, however by the theory about the Todd-Coxeter method it is not guaranteed that such a "pure minimal gap strategy" must terminate, even if the index of h in g is finite since the so called Mendelsohn Condition might be violated (see [Neu82]). If you click this button while no gaps of length 1 exist, a warning `There are no gaps of length 1` will appear in red below the coset table. If the process runs out of gaps of length 1 before ng of such gaps have been closed, the process will stop and a warning `There are no more gaps of length 1` will appear in red below the Coset Table. Note that the strategies for **closing** the tables by filling gaps of length 1 that are offered in the menu of the top button `Close` are not "pure minimal gap strategies", they throw in Felsch type definitions if no gaps of length 1 exist (see 5.1.9).

7 ▶ `back to`

In experimenting with various definition sequences you may want to retrace to a certain point in your procedure. This is offered by this button. Upon clicking it a query window (see Section 1.4) is opened that allows you to retrace **to the situation after defining the coset number that you enter into a given field**. The default offered is to retrace by just one step by offering the last-but-one defined coset number. If you enter a **negative** integer, the CE process is retraced **by** a number of definitions that is the absolute value of the negative integer that you had entered.

8 ▶ `sort defs`

Clicking this button will directly apply the "Sorting Definitions" method (see Section 3.4) to the Coset Table. Note that this button is disabled if the tables have not closed. You may sometimes observe during the running of this method that the number `Defined` in the Information Line is decreasing. When the method has come to an end this will be indicated in the Information Line by the (red) text `Tables sorted`.

9 ▶ `short-cut`

As already pointed out in Section 3.5, the number of coincidences encountered before all tables close may strongly depend on the definition sequence (the strategy followed). Once closure has been reached, the definition sequence that has led to the result is kept, and in case coincidences have occurred one may try to use it for finding a shorter definition sequence that also leads to the result.

An algorithm that tries to do this has been developed by Volkmar Felsch. This "Short-cut" method is described in Section 3.5, here we just mention that it goes through various cycles that each may reduce the number of still needed definitions. By clicking the button `short-cut` a query window (see Section 1.4) is opened in which you can specify the number of cycles of the Short-cut method to be performed or can decide to let it run until it terminates having exhausted all its (finitely many) possibilities. This stepwise reduction can be observed in the Information Line (see 4.3) where the number of defined cosets keeps changing, and the number of cycles of the method performed so far is displayed, until the algorithm stops. While the reduction achieved is often quite remarkable it should be understood that no guarantee is given that the absolute minimum of needed definitions will be found. In fact by experimenting with this feature you can see that the final number of definitions after this "pruning" can depend on the sequence of definitions from which you start. Note that you can inspect the definition sequence before and after this pruning using the bottom button `show defs` (5.2.12). Note that the Short-cut method can only be applied after tables have **closed**, if you try to invoke it before this has been achieved, a warning `The tables have not closed` flashes up in red below the Coset Table and nothing happens.

Of the six white buttons five allow you to open further "display" windows (see Section 1.4), in which additional information about the state of the coset enumeration is shown. Note that clicking the same button a second time will close the corresponding display window.

10 ▶ `show rels`

Clicking this button opens a window that shows the "List of Relators" (see 4.4 for operating this). Note that thus it indirectly gives access to the Relation Tables (see Section 4.5).

11 ▶ `show subgrp`

Clicking this button opens a window that shows the List of Subgroup Generators (see 4.6 for operating this). Note that it thus indirectly gives access to the Subgroup Tables (see 4.7). If no subgroup generators are given, that is if you try to enumerate the cosets of the trivial subgroup, no window is opened but the warning `There are no subgroup generators` flashes up in red below the Coset Table.

12 ▶ `show defs`

Clicking this button with the **left** mouse button opens a window that shows the Table of Definitions (see 4.8). Clicking with the **right** mouse button will instead colour in the Coset Table all those entries green in which the respective coset number was defined in the present definition sequence. We hope that seeing the distribution of the definitions in the coset table may be of interest in particular if such a definition sequence has been obtained by use of the Short-cut method (see 5.2.9 and 3.5).

13 ▶ `show gaps`

Clicking this button opens a window showing a list of "representatives" of the equivalence classes of presently known gaps of length 1. See Section 3.2 and Section 4.9 for a description of the structure and meaning of that list as well as for the operations that can be started by clicking in that list. If no gaps of length 1 exist, no window will be opened, but a warning `There are no gaps of length 1` will flash up in red below the Coset table.

14 ▶ `show coincs`

If no coincidences are pending upon clicking this button the warning `There are no pending coincidences` will flash up in red below the Coset Table. Otherwise clicking this bottom button opens a window which displays the text `Pending coincidences` and a list of these. See `The List of Pending Coincidences` (Section 4.10) for a description of the structure and meaning of that list and the operations that can be started by clicking in that list.

The white button

15 ▶ **mark cosets**

> allows one to mark certain cosets (in green colour) in all tables in order to see easily where they occur. Clicking this bottom button opens a query window (see Section 1.4) in which you can enter the coset numbers (separated by commas) that you want to get coloured. Entering an empty sequence of coset numbers to be marked will erase any markings.

There are finally three red buttons, which allow you to leave your present calculation with ITC in various ways.

16 ▶ **clear**

> Clicking this bottom button will clear all the tables from all entries constructed in the coset enumeration, while keeping the group and subgroup that you had entered into GAP, so that you can directly start anew with the same problem. The table size will be reset to the default table size, but otherwise settings defined during the last run, for instance whether coincidence handling is switched on or off, will remain, that is, not automatically be set to their default values. Remember that you can inspect the present settings in the window opened via the menu entry `show current settings` (5.1.8).

17 ▶ **reset**

> This works similarly to `clear` except that all settings will be reset to their default values. Remember again that you can inspect the present settings in the window opened via the menu entry `show current settings` (5.1.8).

18 ▶ **quit**

> This closes all windows and just brings you back to the GAP window. In a new start of ITC all switches will be set to their default values. The group that you had handled will still be available as a GAP object.

6

Examples

Here we give a number of examples to illustrate various features of ITC. Please note that input files for all these are contained with the distribution of ITC.

6.1 A Presentation by A. Cavicchioli

The presentation used in this section as a first example of the use of ITC stems from a paper by A. Cavicchioli (see [Cav86]). We use it here because it has also been used to explain CE in an easy introduction to computational methods for finitely presented groups by the third author and Said Sidki, of which an English translation is available on the GAP Web Pages, see

```
http://www-history.mcs.st-and.ac.uk/~gap/Info/talks.html       .
```

The GAP input to this group will be

```
F := FreeGroup( "a", "b" );
a := F.1; b := F.2;
rels := [ a^-2*b*a*b^-1*a*b, a^3*b^-1*a^-2*b^-1 ];
G := F / rels;
a := G.1; b := G.2;
H := Subgroup( G, [ a ] );
InteractiveTC( G, H );
```

This can be replaced by reading the input file for this presentation:

```
ItcExample( "cav" );
```

Note that, since the generators of the free group were called a and b, these names will be used by ITC.

The ITC Coset Table shown in the Coset Table Window has already entries for $1 * a$ and $1 * a^{-1}$ which stem from the fact that the subgroup is automatically given coset number 1 at the beginning and consequences of this definition, here via the fact that the subgroup is generated by a, are inserted into the Coset Table. You can inspect the Subgroup Table (in this case only one) by clicking the only subgroup generator in the List of Subgroup Generators which you can get by clicking `show subgrp`.

We first try the simplest solution: we click the top button `Close`, and select `close table by Felsch` from its menu. This does indeed work, filling the table with 12 rows. We can read from the Information Line that no coincidences were encountered in this CE (the number `Deleted` is given as 0).

We want to try different strategies next. Rather than starting from scratch, we clear the table of all entries created by the last CE by just clicking the (red) bottom button `clear`.

Without explicitly mentioning this each time we will do the same, or use `reset` (depending whether we also want to keep settings or not) whenever we want to reuse the definition of the group.

Also the three "gaps of length 1" strategies offered in the menu of the top button `Close` will close the table without encountering coincidences. We remember, however, that these really invoke Felsch steps if no gaps of length 1 are available and we want in a moment to investigate if this was the case here.

The "HLT" strategy, however, (even though it uses consequences) needs 13 definitions, i. e. produces one coincidence.

That is, we have in this case ended with a slightly non-optimal definition sequence (which we can inspect either in a separate window or by marking the definitions green in the Coset Table by clicking the (white) `show defs` bottom button with the left or the right mouse button, respectively). We try next, if the Short-cut method will be able to improve this definition sequence. Pressing the (green) `short-cut` bottom button and not restricting the number of loops in the query window does in fact produce in five loops a definition sequence in which no coset has to be removed.

Prescribing in a repetition of the previous experiment in the Short-cut query window only one loop shows us that already this one loop suffices to produce a definition sequence closing the tables without coincidences.

We next want to follow some of these computations stepwise.

First we try the Felsch method stepwise. Clicking the (green) bottom button `Felsch` opens a query window, in which we can enter how many Felsch steps we want to perform. The value 1 is offered, and we should in fact use this to watch how by repeated call of one Felsch step at a time the tables close. It will be informative, not only to watch what happens in the Coset Table, but also to watch what happens in the Relation Tables. For this purpose we first click the (white) bottom button `show rels` which opens a window showing the two relators. Clicking these opens two more windows displaying the Relation Tables. After each step the new entries will be shown in red. While after each of the definitions of coset number 2 and 3 just two new entries in the coset table are made, e. g. in the first case corresponding to the definition $2 = 1 * b$ and the trivial consequence $2 * b^{-1} = 1$, after definition of coset number 4 we observe that for the first time rows in the relation tables close and yield further consequences and hence entries into the coset table. Again after having defined coset number 12 the tables will be closed without a coincidence having occurred (and this is indicated in the Information Line).

We may repeat a Felsch strategy but with larger steps, that we enter into the field of the query window that opens after clicking the green bottom button `Felsch`.

We now want to return to the question what really happens when we use a "gaps of length 1" strategy offered for closing a table:

If at the start we click `show gaps`, we get the message `There are no gaps of length 1` and if we like, we can (partially, compare 5.2.13!) confirm this by looking at the Subgroup Table via the bottom button `show subgrp` (which turns out to be closed already) and the two Relation Tables via the bottom button `show rels` which in fact each have only one row yet with gaps of length 4 and 3, respectively.

After performing a first Felsch step (using the bottom button `Felsch`) the situation has not improved, there are still no gaps of length 1, however after a further Felsch step a first gap of length 1 turns up. After filling this, no new gap of length 1 turns up, so we again resort to a Felsch step, after which we can again fill a gap of length 1. One more Felsch step then allows us to use gap of length 1 filling four times after which a last Felsch step is needed to close the table.

We will now try what happens if we follow a Haselgrove-Leech-Trotter kind of strategy. We can do this in two ways:

We can use the bottom button `HLT` to make a number of definitions (that we can prescribe) following the HLT strategy. If we do this, and make one definition at a time, we observe that with defining coset number 5 we have produced a coincidence and coset number 4 will get eliminated.

We can also close rows one by one. For this we press the (green) bottom button `fill rows`. In a query window we are asked which row we want to fill (in all – in this case, two – Relation Tables). We choose the proposed default value, in the first instance, so closing the first rows. This already defines five coset numbers. However, if we watch the Information Line carefully, it tells us that in fact one of these five defined coset numbers has already been eliminated again by a coincidence. The Coset Table shows that this has been coset 4 (and no wonder, we have followed the same definition sequence as in our last experiment).

If we apply a few more steps of a strategy which consists of just clicking the button `fill rows` and then accepting the row number proposed by the ITC, the next proposal will be to close row 2 in each table. Doing this brings us already to coset number 10. Then, as expected, we are offered to fill the third rows. This step does not only lead to another definition, namely of coset number 11, but also implies that all rows up to the seventh are already closed (you may easily check this by looking at the Relation Tables). Hence, in the next step the proposal is to fill the eighth rows and doing this closes the tables. In these tables no coset number 4 occurs, the coset numbers run from 1 to 13 with that omission.

We may next want to see a bit closer what happens with that coincidence, and we can do so by pulling down the `Settings` menu from the top button `Settings` and releasing the mouse button on the menu entry `coincidence handling off`. If now we repeat the same procedure as last time, namely calling `fill rows`, the process will come to a halt, while the coset number 4 is still "alive". ITC warns us in the Information Line that coincidences are pending and offers to open a window showing the coincidence(s) pending, in this case $4 = 3$. We can eliminate coset number 4 by clicking this equation in the window, after which the warning vanishes and we have the state that in the previous run we had after the first `fill rows` command. We can get to the end in the same way again.

We could vary the HLT procedure by not always filling a particular row in both Relation Tables simultaneously, but in only one of these tables. This we can achieve by clicking the first or last entry in the chosen row of that Relation Table. In fact, if we just proceed by closing rows of the first Relation Table we will again get closure after one coincidence has occurred, which however can again be elimintated by the Short-cut method, while proceeding by only closing rows in the second Relation Table, we even arrive at the result without coincidences.

We have seen at the beginning that a Felsch strategy brought us to the end without having encountered coincidences. Next we show how delicate all these statements are. We pretend that the columns of the Coset Table have been reordered so that first both generators and then their inverses head the columns of the Coset Table. Since we cannot change the program to follow a Felsch strategy with this arrangement of columns, we do it by hand clicking consecutively the following points after having switched coincidence handling off:

$$1 * b, 1 * b^{-1}, 2 * a, 2 * b, 2 * a^{-1}, 3 * a, 3 * b^{-1}, 5 * a, 5 * b, 5 * a^{-1}.$$

After this one we get a coincidence $9 = 8$; we remove 9 by clicking this equation in the window displaying it, but this produces immediately a further coincidence $10 = 8$, only after also removing 10 we can continue

$$7 * b^{-1}, 11 * b, 11 * a^{-1}$$

which leads to closure of the tables. Thus we have encountered even two coincidences and at a different place. `short-cut` will again yield a definition sequence without coincidences in only one loop.

Next let us just observe that we can in fact produce even more coincidences by using a "stupid" definition sequence. We may e. g. first define consecutively 10 images under b and then 10 further ones under a, without any indication of the table closing. If we next perform 3 Felsch steps, we get a first coincidence. Eliminating this will produce a further one and so on, and if we continue (always clicking the first one of the displayed coincidences) we will see that at certain points half a dozen coincidences are pending simultaneously. Working these off, we will eventually arrive at a table with no further coincidences pending, but which will still not have closed. After two more Felsch steps we reach closure again, but the definition sequence will list 26 coset numbers. Short-cut will again reduce this to a definition sequence of 12 coset numbers but if we follow what happens loop by loop, we see that only the fifth loop brings a reduction to 20 coset numbers, while only the eighth (and last) loop brings the reduction to a definition sequence without coincidences.

Finally we give an example in which `short-cut` will not completely succeed. We first define "by hand" four images under b^{-1}:

$$1 * b^{-1}, 2 * b^{-1}, 3 * b^{-1}, 4 * b^{-1},$$

and then close the table using the Felsch strategy. The table will close after having gone through two coincidences. If we now invoke the Short-cut method, it will reduce the definition sequence from 14 to 13 definitions, but not to 12. However in this case `sort defs` will bring us down to a definition sequence of 12 definitions either applied to the definition sequence of 14 definitions obtained originally or to the "pruned" definition sequence of 13 definitions.

In this first example we have already seen some of the multitude of ways in which we can modify the basic idea of CE. In order to demonstrate more of the functionality of ITC, we will study further examples.

6.2 The Fibonacci Group F(2,7)

The Fibonacci Group F(2,7) is cyclic of order 29, but this result is not easily obtained. The first and crucial step is to establish that the group is in fact cyclic by showing with the help of coset enumeration that one of the cyclic subgroups generated by one of the generators has index 1 in F(2,7). In her beautifully written Master's thesis [Ede89] Margaret Edeson describes the history of that proof and uses it to discuss in detail approaches to the problem of finding a short definition sequence that will lead to the result that this index is trivial.

In fact she documents carefully correspondence around that problem, involving in particular John Leech and George Havas. By 1984 John Leech actually had obtained a definition sequence of 55 definitions by a posteriori pruning (by hand) much longer definition sequences determined a priori. Again by elaborate hand work in 1989 Margaret Edeson was able to get this number down to 53. To read her comments about at various stages feeling unsafe whether to continue enumerations because of the fear of already having made a mistake is not only amusing but an encouragement for developing programs such as ITC.

The GAP input for the enumeration of the cosets of the cyclic subgroup generated by the third generator reads:

```
F := FreeGroup( "a", "b", "c", "d", "e", "f", "g" );
a := F.1; b := F.2; c := F.3; d := F.4; e := F.5; f := F.6; g := F.7;
rels := [
  a*b*c^-1,
  b*c*d^-1,
  c*d*e^-1,
  d*e*f^-1,
  e*f*g^-1,
  f*g*a^-1,
  g*a*b^-1
];
G := F / rels;
c := G.3;
H := Subgroup( G, [ c ] );
InteractiveTC( G, H );
```

This can be replaced by reading the input file for this presentation:

```
ItcExample( "f27" );
```

Note that again ITC will print the generators of the free group using their names a, b, etc.

The Coset Table shown by ITC has already the entries for $1 * c$ and $1 * c^{-1}$. Moreover we note that it already indicates gaps of length 1 by (red - meaning new) dots in the Coset Table which stem from the fact that the relations are very short.

We first click the top button `Close` and choose the menu entry `close table by Felsch`. Rather quickly we get closure with index 1 after 332 definitions. `Deleted: 331` confirms that in fact 331 coincidences have been removed. It will be instructive to follow this coincidence handling step by step for a few steps.

We can do this by returning to the beginning using the bottom button `clear` and then selecting the menu entry `coincidence handling off` in the menu of the top button `Settings`. Starting now `close table by Felsch` again, the enumeration will stop with 332 coset numbers defined and the warning `Pending coincidences` in the Information Line. At the same time a window for the Table of Pending Coincidences is offered and this informs us that at present just one coincidence, namely 226 = 106 has been found.

We can use the scroll buttons to inspect the row 226 that will be removed by handling this coincidence. We click the bottom button `scroll to` and write 226 instead of the number of the last row (332) into the field offered, then clicking `ok` scrolls the Coset Table Window such that row 226 occurs in the middle of the window. We see that row 226 is coloured red (while row 106 remains black).

We can now click, in the List of Pending Coincidences, on that coincidence 226 = 106 and this will be eliminated. Looking again for row 226 in the Coset table (or any of the Relation Tables) we will not find it any more, it has got suppressed, but no renumbering has taken place, row 227 now follows immediately after row 225.

However the warning `Pending coincidences` has not vanished and indeed the window with the List of Pending Coincidences now shows us two new coincidences: 124 = 43 and 229 = 21 that have now been detected. We can repeat the same game and see that for some time the List of Pending Coincidences grows longer and longer until (if we have the patience to follow that game to its end instead of just switching on the automatic coincidence handling again) indeed all coincidences are worked off and index 1 is obtained. That is, we see in this case how one single coincidence triggers the "collapse" of the Coset Table.

Invoking now the Short-cut method for the definition sequence of length 332 not prescribing the number of loops, after 33 loops it stops with a definition sequence of 54 definitions (already one better than Leech's result by hand). The Short-cut method is slow enough to be able to see in the Information Line how the number of definitions is brought down in several of the loops of the Short-cut method. It will be instructive to do the 33 loops one by one, to see that not all of them really improve the definition sequence. In fact at first there are some dramatic improvements of the original number of 332 definitions as shown by the number of definitions obtained in the first six loops: 327, 251, 226, 145, 132, 127. But then the next loop brings no improvement, and later on there are six consecutive loops during which the number of definitions stays at 64 definitions. By the way, if you do this interactively step by step, watch the counter for the number of loops, if this does not rise any more, the method will have come to its natural end. Also the displayed text changes: While it says i `Short-cut loops` as long as the Short-cut method works, it changes to `Short-cut` (i `loops`) after the Short-cut method has stopped.

We next choose the `use gaps strategy 2 (first rep of max weight)` entry in the menu of the top button `Close`. We are rewarded by obtaining a priori (!) a definition sequence of only 126 definitions leading to collapse of the table (which is better than anything Leech had at his disposal). But if we expected that now also `short-cut` would beat records, we are disappointed, it stops after 35 loops but still with a definition sequence of length 57.

So finally we choose the `use gaps strategy 3 (last rep of max weight)`. This closes after only 79 cosets defined, `short-cut` gets down to 54 in only 29 loops. However if we first perform three Felsch steps (using the bottom Button `Felsch` and prescribing 3 in the field provided in the query window) and then close again by `use gaps strategy 3` we get once more closure after 79 definitions, but this time `short-cut` brings us down to a definition sequence of length 50 (breaking all presently known records).

Of course this has not been our main goal, but rather to demonstrate the flexibility of using ITC. In fact for somebody interested in CE it will be very fascinating to study this example further: if you use n Felsch steps and then close by `use gaps strategy 3` and follow this by the Short-cut method, the series of results will be most puzzling. We do not want to betray it completely here - just try it out - but we mention that for $n = 20$ we obtained an enumeration that we did not even finish because it went on and on (we did follow it until it had defined more than 10000 coset numbers), while for $n = 40$ we were at that very good situation of definition sequence length 79 a priori and 50 a posteriori again.

Just for comparison: `close table by HLT with consequences` needs 2276 definitions.

Finally we mention that there are simple ITC strategies to find definition sequences of length 50 also if we replace the subgroup $H = \langle c \rangle$ by any of the other cyclic subgroups generated by the generators of G. In each of these cases it suffices to start with one or two suitable ITC commands and then again to close the tables via `use gaps strategy 3` and to apply `short-cut`. An appropriate start may e. g. consist of doing 10, 1, 13, or 10 Felsch steps in case $H = \langle a \rangle$, $H = \langle d \rangle$, $H = \langle f \rangle$, or $H = \langle g \rangle$, respectively, or of clicking $1 * f$ or $1 * a$ and $1 * b$ in the Coset Table for $H = \langle b \rangle$ or $H = \langle e \rangle$, respectively.

6.3 A Presentation by B.H. Neumann

Bernhard Neumann (see e. g. [Neu79]) has discussed a famous sequence of presentations of the trivial group of increasing difficulty for any CE method. In fact for a long time the only one that could be done was the first of them that we consider here, (called $e1$ here), the next one was only recently "cracked" by George Havas' and Colin Ramsey's program ACE.

The GAP input for the presentation is

```
F := FreeGroup( "a", "b", "c" );
a := F.1; b := F.2; c := F.3;
rels := [ c^-1*a*c*a^-2, a^-1*b*a*b^-2, b^-1*c*b*c^-2 ];
G := F / rels;
H := TrivialSubgroup( G );
InteractiveTC( G, H );
```

This can be replaced by the input file for this presentation:

```
ItcExample( "e1" );
```

We just mention a few results, but mainly suggest this presentation for further exercises:

`close table by Felsch` needs 588 definitions, `short-cut` reduces to 68 in 31 loops.

`use gaps strategy 1` needs 119 definitions, `short-cut` reduces to 68 again this time in 33 loops.

`use gaps strategy 2` needs 116 definitions, `short-cut` reduces to 68 again in 33 loops.

`use gaps strategy 3` needs 119 definitions, but `short-cut` reduces to 64 in 27 loops, which coincides with what Havas and Ramsey got by hand pruning a somewhat better a priori enumeration with only 81 definitions obtained by experimenting with ACE (see [HR99]).

However using 13 Felsch steps followed by closing by `use gaps strategy 1` yields a definition sequence of length 134 that is reduced to only 61 by the Short-cut method in 30 loops. The same procedure with 196 Felsch steps and closing by `use gaps strategy 3` gets even down to a sequence of 60 definitions.

By the way this presentation also provides an example of a rather strong reduction of the definition sequence by `sort defs`: Using `close table by HLT with consequences` produces a definition sequence of 672 coset numbers that reduces to 279 under `sort defs`.

6.4 The Group (8,7;2,3)

As a final example we take the presentation and subgroup described by the GAP input:

```
F := FreeGroup( "a", "b" );
a := F.1; b := F.2;
rels := [ a^8, b^7, (a*b)^2, (a^-1*b)^3 ];
G := F / rels;
a := G.1; b := G.2;
H := Subgroup( G, [ a^2, a^-1*b ] );
InteractiveTC( G, H );
```

This can again be replaced by reading it from a file:

```
ItcExample( "g8723" );
```

This presentation has also been used frequently for comparisons of CE strategies, e. g. in [CDHW73].

If we start with all parameters at default values and choose `close table by Felsch` the coset enumeration will be interrupted after the definition of 1000 coset numbers and a window springs up in which it is offered to `extend table size from 1000 to` and then in a field the number 2000 is proposed (that can be changed). After clicking `ok` the requested extension of the table size is done and the enumeration resumed. If one clicks `cancel` instead, thus rejecting the offer, one gets the warning `Insufficient table size` in red below the Coset Table, and then has to decide if one wants to extend the table size (now using the menu entry of the top button `Settings`) or to give up this enumeration.

The enumeration stops having defined 1306 coset numbers, but the index has turned out to be 448. `short-cut` reduces this rather redundant sequence of coset numbers to 472 in 140 loops (but this already takes a little while).

Choosing `use gaps strategy 1` from the menu of `Close` needs 825 definitions which boil down to 470 under `short-cut` in 141 loops.

For comparison we mention that the shortest definition sequence found by George Havas and Colin Ramsay experimenting with **a priori** methods using ACE has length 662 (see [HR01]).

Bibliography

[Ban56]   P. A. Bandler. A method for enumerating the cosets of an abstract group on a digital computer. M. A. thesis, University of Manchester, 1956.

[Cav86]   A. Cavicchioli. A countable class of non-homeomorphic homology spheres with Heegard genus 2. *Geometria Dedicata*, 20:345–348, 1986.

[CDHW73]   J. J. Cannon, L. A. Dimino, G. Havas, and J. M. Watson. Implementation and analysis of the Todd-Coxeter algorithm. *Mathematics of Computation*, 27:463–490, 1973.

[Ede89]   M. Edeson. Investigations in coset enumeration. M. A. thesis, Canberra College of Advanced Education, since 1990 University of Canberra, 1989.

[Fel61]   H. Felsch. Programmierung der Restklassenabzählung einer Gruppe nach Untergruppen. *Numerische Mathematik*, 3:250–256, 1961.

[Hav91]   George Havas. Coset enumeration strategies. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation (ISSAC'91), Bonn 1991*, pages 191–199. ACM Press, 1991.

[HR99]   G. Havas and C. Ramsay. The trivial group made easy, a case study in coset enumeration. Draft Manuscript, 1999.

[HR01]   G. Havas and C. Ramsay. Experiments in coset enumeration. 2001.

[Lee84]   J. Leech. Coset enumeration. In Michael D. Atkinson, editor, *Computational Group Theory, Proceedings LMS Symposium on Computational Group Theory, Durham 1982*, pages 3–18. Academic Press, 1984.

[Neu79]   B. H. Neumann. Proofs. *The mathematical intelligencer*, 2:18–19, 1979.

[Neu82]   Joachim Neubüser. An elementary introduction to coset table methods in computational group theory. In Colin M. Campbell and Edmund F. Robertson, editors, *Groups-St.Andrews 1981, Proceedings of a conference, St.Andrews 1981*, volume 71 of *London Math. Soc. Lecture Note Series*, pages 1–45. Cambridge University Press, 1982.

[Ram99]   C. Ramsay. *ACE User Manual*. Department of Computer Science & Electrical Engineering and Department of Mathematics, The University of Queensland, QLD 4072, Australia, 1999. Draft.

[Sim94]   C. C. Sims. *Computation with Finitely Presented Groups*. Cambridge University Press, 1994.

Index

This index covers only this manual. A page number in *italics* refers to a whole section which is devoted to the indexed subject. Keywords are sorted with case and spaces ignored, e.g., "PermutationCharacter" comes before "permutation group".