J. Symbolic Computation (1995) 19, 459–488

The Todd–Coxeter Procedure and Left Kan Extensions

S. CARMODY, M. LEEMING AND R. F. C. WALTERS †

Department of Pure Mathematics, University of Sydney, Sydney, NSW 2006, Australia

(Received 27 June 1991)

We introduce a generalization of the Todd–Coxeter procedure for the enumeration of cosets. The generalized procedure relates to a construction in category theory known as the left Kan extension. It admits of a great variety of applications, including enumerating cosets, computing certain colimits in the category of **Sets**, and enumerating the arrows in a category given by generators and relations.

We begin by defining the notion of a left Kan extension, and giving a number of illustrative examples. We then provide a full specification of the procedure, followed by its application in relation to each of the examples. Finally, we provide a formulation of the procedure in terms of graphs and presentations of actions of graphs (automata) which is more convenient for theoretical purposes.

1. Introduction

The *Todd–Coxeter procedure* for the enumeration of cosets, as described in Coxeter and Moser (1957), Mendelsohn (1964), Leech (1984), Johnson (1990) and Sims (1994), has long been a powerful tool in computational group theory. The underlying procedure is, however, not specific to group theory and appears in various guises in a number of other areas of computational algebra. Here we describe a generalization of the procedure which relates to a construction in category theory known as the *left Kan extension*. The enumeration of cosets then becomes a special case of this generalized procedure, but the general procedure may also be used for a great variety of other purposes. Some examples include listing the arrows of a category given generators and relations, and computing colimits in the category of **Sets** such as the orbits of a group action and the coequalizer of two functions.

When considering the Todd–Coxeter procedure and other important procedures of computational algebra such as the Knuth–Bendix procedure, placing them in the context of category theory serves at least two purposes. First, they can then be useful as tools for category theorists in much the same way as the original Todd–Coxeter procedure was for group theorists (an early paper in this direction is Mora, 1978). Second, it is our contention that the language of category theory is a valuable one for unifying and clarifying the concepts behind these powerful procedures. The significance of attempting such a unification is illustrated in Buchberger (1987) where the similarities of the

† The authors gratefully acknowledge the support of the Australian Research Council.

0747–7171/95/050459 + 30 $08.00/0

Knuth–Bendix procedure and its relatives are explored. When passing from categories to categories with structure, the language becomes even more expressive. In a future publication, we shall describe a further extension of the Todd–Coxeter procedure to categories with products (see Leeming and Walters, 1995). With this extension one is able compute left adjoints to algebraic functors which allows for example the enumeration of elements of any finitely-presentable multisorted algebra.

We begin here in Section 2 by defining the notion of a left Kan extension and give some examples which are pertinent to the generalized Todd–Coxeter procedure which is described in Section 3. In Section 4 a full specification of the procedure is given followed by a series of illustrative examples of its use in different cases in Section 5.

For ease of exposition of the general procedure and the worked examples, we adopt the somewhat old-fashioned approach of using tables. These are, of course, not appropriate for some theoretical purposes, and so in Section 6 we briefly discuss a formulation in terms of graphs and presentations of actions of graphs (automata). With this formulation, one may address proofs of correctness and termination. This is done in Carmody and Walters (1990) for the crucial sub-procedure known as *dealing with coincidences*.

2. Left Kan extensions

To begin, we establish the context for the procedure using some category theory. The terminology involved is all described in MacLane (1971); chapter 10 in particular deals with Kan extensions.

Central to the procedure are functors of the form $\mathbf{C} \to \mathbf{Sets}$ from a (small) category $\mathbf{C}$ into the category of $\mathbf{Sets}$; we also refer to such functors as *actions* of the category $\mathbf{C}$. The aptness of this name lies in the fact that we can consider a group as a category with one object whose arrows are the group elements (composition corresponding to multiplication). Then a functor into $\mathbf{Sets}$ corresponds simply to the usual notion of group action.

Given an action $X: \mathbf{A} \to \mathbf{Sets}$ of a category $\mathbf{A}$ and a functor $F: \mathbf{A} \to \mathbf{B}$ into another category $\mathbf{B}$, the left Kan extension, which is defined precisely below, is the action of $\mathbf{B}$ which is the universal extension of the action X along the functor F. To see the relevance of left Kan extensions to the Todd–Coxeter procedure, given a group G with a subgroup H, consider the following diagram:

$$\begin{tikzcd}
\mathbf{H} \arrow[rr, "\subseteq"] \arrow[dr, "*"'] & & \mathbf{G} \arrow[dl, "G/H"] \\
& \mathbf{Sets} &
\end{tikzcd}$$

where $*$ represents the trivial action of H on a one-point set and G/H represents the usual action of G on the (left) cosets of H by (left) multiplication. Along the top of the diagram is the obvious inclusion functor. In a sense made precise by the notion of left Kan extension, the action G/H is the extension of $*$ along this inclusion.

We now give the definition of the left Kan extension of an action.

DEFINITION 2.1. *Given functors $F: \mathbf{A} \to \mathbf{B}$ and $X: \mathbf{A} \to \mathbf{Sets}$, a left Kan extension of X along F consists of a functor $L: \mathbf{B} \to \mathbf{Sets}$ (sometimes denoted $\mathrm{Lan}_F X$) and a natural transformation $\epsilon: X \to LF$ with the universal property that for each $M: \mathbf{B} \to \mathbf{Sets}$ and $\alpha: X \to MF$, α factors through ϵ as $\alpha = \sigma F \cdot \epsilon$ for a unique $\sigma: L \to M$.*

Diagrammatically:

$$
\begin{tikzcd}
\mathbf{A} \arrow[rr, "F"] \arrow[dr, "X"'] & \arrow[d, phantom, "\overset{\varepsilon}{\Longrightarrow}"] & \mathbf{B} \arrow[dl, "L"] \arrow[dl, bend left, "M"] \arrow[dl, phantom, bend left=20, "\overset{\Longrightarrow}{\sigma}"] \\
& \mathbf{Sets} &
\end{tikzcd}
\quad = \quad
\begin{tikzcd}
\mathbf{A} \arrow[rr, "F"] \arrow[dr, "X"'] & \arrow[d, phantom, "\overset{\alpha}{\Longrightarrow}"] & \mathbf{B} \arrow[dl, "M"] \\
& \mathbf{Sets} &
\end{tikzcd}
$$

For completeness, we now give an explicit formulation of the left Kan extension for this case, since it helped provide the insight leading to the development of the procedure.

THEOREM 2.1. *Given functors $F: \mathbf{A} \to \mathbf{B}$ and $X: \mathbf{A} \to \mathbf{Sets}$, where $\mathbf{A}$ and $\mathbf{B}$ are small categories, the left Kan extension, L, of X along F is given as follows:*

(i) on objects

$$LB = \Big(\sum_{A \in \mathbf{A}} \mathbf{B}(FA, B) \times XA\Big)\Big/\!\sim \qquad \textit{for all } B \in \mathbf{B},$$

where $\sim$ is the smallest equivalence relation such that for all $f: A \to A'$ in $\mathbf{A}$, $g: FA' \to B$ in $\mathbf{B}$, $x \in XA$,

$$(gFf, x) \sim (g, Xf(x)); \tag{2.1}$$

(ii) on arrows

$$Lh: LB \to LB': [g, x] \longmapsto [hg, x] \qquad \textit{for all } h: B \to B' \text{ in } \mathbf{B}$$

where we denote the equivalence class of (g, x) by $[g, x]$.

PROOF. This is a special case of the coend formula:

$$LB = \int^{A} \mathbf{B}(FA, B) \cdot XA \qquad \text{for all } B \in \mathbf{B}$$

in theorem 1 of section 10.4 in MacLane (1971). However, in this case it is straightforward to establish the result directly.

The important step is to define

$$\epsilon_A: XA \to LFA: x \longmapsto [1_{FA}, x].$$

Setting $g = 1_{FA'}$ in equation (2.1) then gives:

for all $f: A \to A'$ in $\mathbf{A}$, $x \in XA$,

$$\begin{aligned}
[Ff, x] &= [1_{FA'}, Xf(x)] \\
\text{so} \quad LFf \cdot \epsilon_A(x) &= \epsilon_{A'} Xf(x)
\end{aligned}$$

and hence $\epsilon: X \to LF$ is a natural transformation. It is then routine to check the details: that L is functorial, that it satisfies the appropriate universal condition and so on. The essential idea is that equation (2.1) is required to force ϵ to be a natural transformation; this naturality condition is made quite explicit in the general Todd–Coxeter procedure. In the context of cosets of a group, for example, the naturality condition corresponds to requiring that the G action be trivial when restricted to H.

Note: $\mathbf{A}$ and $\mathbf{B}$ are required to be small so that the sum in the expression for L is actually meaningful. □

We conclude this section with a definition which is needed to discuss the termination of the procedure.

DEFINITION 2.2. *A functor* $M: \mathbf{C} \to \mathbf{Sets}$ *is finite if for all* $C \in \mathbf{C}$, MC *is finite.*

The procedure will in fact terminate for all cases in which the the left Kan extension, L, is finite.

3. Examples of left Kan extensions

We now give some examples of left Kan extensions, which will later provide illustration of the use of the procedure. In these examples, we consider the elements of monoids or groups as the arrows of a one-object category (this object is usually referred to as $*$), with composition corresponding to multiplication. We also often refer to the category $\mathbf{1}$, which has one object, denoted $\bullet$, and only an identity arrow.

(i) *Monoids*

Consider the following situation:

$$\begin{tikzcd}
\mathbf{1} \arrow[rr, "\subseteq"] \arrow[dr, "X"'] & & \mathbf{M} \arrow[dl, "L"] \\
& \mathbf{Sets} &
\end{tikzcd}$$

where M is a monoid and $X\bullet$ is a one-point set. In this case the left Kan extension simply gives the elements of the monoid. That is:

$$L* = \{m \in M\}$$

and the effect on arrows is premultiplication, that is:

$$Lm: M \to M : x \longmapsto mx \qquad \forall m \in M.$$

Also, ϵ_* maps the one-point set to the identity of M.

(ii) *Categories*
Consider

$$\begin{tikzcd} \mathrm{obj}\,\mathbf{C} \arrow[rr, "\subseteq"] \arrow[dr, "X"'] & & \mathbf{C} \arrow[dl, "L"] \\ & \mathbf{Sets} & \end{tikzcd}$$

where obj $\mathbf{C}$ is a category consisting of the objects of $\mathbf{C}$ with only identity arrows, and X maps all the objects to one-point sets. Here the left Kan extension gives the arrows of the category $\mathbf{C}$. More precisely,

$$L = \sum_{C' \in \mathbf{C}} \mathbf{C}(C', \_).$$

This clearly comes directly from Theorem 2.1. Here ϵ maps the one-point sets to the identity arrows. In the case when $\mathbf{C}$ only has one object, this example reduces to (i).

(iii) *Cosets of a group*
Consider

$$\begin{tikzcd} \mathbf{H} \arrow[rr, "\subseteq"] \arrow[dr, "X"'] & & \mathbf{G} \arrow[dl, "L"] \\ & \mathbf{Sets} & \end{tikzcd}$$

where H is a subgroup of the group G and $X*$ is a one-point set. Here the left Kan extension gives the cosets of H. That is, $L* = G/H$, and

$$Lg: G/H \to G/H : xH \longmapsto gxH.$$

This follows from Theorem 2.1, since the expression for $L*$ reduces to $G/\sim$ and equation (2.1) to $gh \sim g$ for all $h \in H$. However,

$$\begin{aligned} g_1H = g_2H \quad &\iff \quad g_2^{-1}g_1 \in H \\ &\iff \quad g_2^{-1}g_1 = h \quad \text{for some } h \in H \\ &\iff \quad g_1 = g_2h \quad \text{for some } h \in H. \end{aligned}$$

Thus the partioning of G into cosets is in this instance precisely the equivalence relation expressed by equation (2.1). Also, ϵ_* maps the one-point set to the coset H

itself. It is this example which provides the connection to the original Todd–Coxeter procedure.

(iv) *Orbits of a group action*
Consider

$$\begin{tikzcd} \mathbf{G} \arrow[rr, "!"] \arrow[dr, "X"'] & & \mathbf{1} \arrow[dl, "L"] \\ & \mathbf{Sets} & \end{tikzcd}$$

where $!: \mathbf{G} \to \mathbf{1}$ is the obvious trivial functor. In this case, X is simply a group action. That is, if $X* = \Omega$, then we have an action $G \times \Omega \to \Omega: (g, x) \longmapsto Xg(x)$. Applying Theorem 2.1 again, $L = X*/\sim$ and equation (2.1) is given by $x \sim Xg(x)$ for all $g \in G$ and $x \in X*$, that is to say $L = \Omega/\sim$ with $x \sim g \cdot x$. This equivalence relation simply partitions Ω into the orbits of the group action. Thus the left Kan extension gives $L\bullet = \Omega/G$, and ϵ_* maps each element of Ω to its orbit.

(v) *Coequalizers in* **Sets**
Consider

$$\begin{tikzcd} \bullet \rightrightarrows \bullet \arrow[rr, "!"] \arrow[dr, "X"'] & & \mathbf{1} \arrow[dl, "L"] \\ & \mathbf{Sets} & \end{tikzcd}$$

where $\bullet \rightrightarrows \bullet$ represents the category

$$\begin{tikzcd} A \arrow[r, bend left, "f"] \arrow[r, bend right, "g"'] & B \end{tikzcd}$$

In this case, using an argument similar to that of (iv), the left Kan extension gives the coequalizer of the two functions Xf and Xg in **Sets**. This can be expressed explicitly as

$$L\bullet = XB/\sim,$$

where $\sim$ is the smallest equivalence relation such that

$$Xf(x) \sim Xg(x) \qquad \text{for all } x \in XA.$$

Then ϵ_B is the canonical projection on to quotient and $\epsilon_A = \epsilon_B Xf = \epsilon_B Xg$. In fact all colimits in **Sets** are examples of left Kan extensions.

4. The general procedure

In this section, the generalized Todd–Coxeter procedure is described. In its full generality, the description becomes somewhat complicated, since there is a substantial amount of information to contend with. Therefore in Section 5 a variety of worked calculations are described for simple cases so as to clarify the processes involved. A more formal algorithm for the procedure is contained in the Appendix.

4.1. SPECIFICATION

We now specify precisely the input data that the generalized Todd–Coxeter procedure requires, and the resulting output it produces if it terminates.

Input

Category $\mathbf{A}$: objects and generating arrows; i.e. a finite graph G_A.

Category $\mathbf{B}$: objects and generating arrows; i.e. a finite graph G_B.

Relations $\mathcal{R}$: a set of "relations" between the generating arrows of $\mathbf{B}$. More precisely, a finite set of pairs of arrows with matching domain and codomain in the free category $\mathcal{F}G_B$ on the generators of $\mathbf{B}$. If $(f_1, f_2) \in \mathcal{R}$, we say $f_1 = f_2$ in $\mathcal{R}$.

Functor X: a collection of finite sets XA, for each $A \in \mathbf{A}$ (for simplicity, these sets will always be specified as $[n] = \{1, \ldots, n\}$), and functions $Xf: XA_1 \to XA_2$, $\forall$ generators $f: A_1 \to A_2$ in $\mathbf{A}$.

Functor F: a map from the objects of $\mathbf{A}$ to the objects of $\mathbf{B}$, and a map from the arrows of G_A to the arrows of $\mathcal{F}G_B$; that is, each generator of $\mathbf{A}$ maps to a path in the generators of $\mathbf{B}$.

Output

Left Kan extension $L: \mathbf{B} \to \mathbf{Sets}$
(given on generating arrows)

Natural transformation $\epsilon: X \to LF$

The procedure will terminate if and only if L is finite.

4.2. TERMINOLOGY

4.2.1. TABLES

The following tables extend the basic multiplication and relation tables of the traditional Todd–Coxeter algorithm. In the general case, there are four types of tables. The first two types contain information on sets and partial functions which are generated during the course of the procedure, while the last two types are used to impose relations between the elements of the sets generated. We describe the tables as they are at the beginning of the algorithm, initialized with all the input data.

(i) ϵ-tables. For all $A \in \mathbf{A}$ there are tables of the form

ϵ_A

$XA \to LFA$	
1	
2	
$\vdots$	
n	

where in this case, $XA = [n]$. When the procedure terminates these tables will give the components of the natural transformation ϵ.

(ii) L-tables. These tables are perhaps best thought of as a collection of tables for each $B \in \mathbf{B}$, generators $g_i: B \to B_i$ in $\mathbf{B}$

Lg_i

$LB \to LB_i$	

But for convenience, we will compress these to the following: for all $B \in \mathbf{B}$

	Lg_1	Lg_2		Lg_n
LB	LB_1	LB_2	$\cdots$	LB_n

where $\{g_1, \ldots, g_n\} = \{g \text{ in } \mathbf{B} \mid \operatorname{dom} g = B\}$. When the procedure terminates, these tables give the effect of the functor L on objects and generating arrows of $\mathbf{B}$.

(iii) Relation tables. For each relation $g_n \ldots g_1 = h_m \ldots h_1: B_1 \to B_2$ in $\mathcal{R}$,

	Lg_1	Lg_n		Lh_1	Lh_m
$LB_1 \to \cdots \to LB_2$			$LB_1 \to \cdots \to LB_2$		
	$\cdots$			$\cdots$	

(iv) Naturality tables. For all generators $f: A_1 \to A_2$ in $\mathbf{A}$, if $Ff = g_n \cdots g_1$,

$XA_1 \xrightarrow{Xf} XA_2 \xrightarrow{\epsilon_{A_2}} LFA_2$			$XA_1 \xrightarrow{\epsilon_{A_1}} LFA_1 \xrightarrow{Lg_1} \cdots \xrightarrow{Lg_n} LFA_2$			
1	5		1			
2	3		2		$\cdots$	
$\vdots$	$\vdots$		$\vdots$			
n	2		n			

where for the purpose of illustration, we have taken $XA_1 = [n]$ and $Xf\colon 1 \mapsto 5$, $2 \mapsto 3, \ldots, n \mapsto 2$.

Note: Tables (i) and (iv) will be of fixed length, determined by the sizes of the XA, whilst in general, Tables (ii) and (iii) will vary in length over the course of the algorithm.

4.2.2. UNDEFINED ELEMENTS

Undefined elements are unfilled entries are in Tables (i) or (ii) in rows which have elements appearing in the left-hand column. For example:

$XA \xrightarrow{\epsilon_A} LFA$	
1	$\square$
2	
$\vdots$	
n	

or

LB	LB_1	LB_2	$\cdots$	LB_n
	Lg_1	Lg_2		Lg_n
1	3	4	$\cdots$	7
2	$\square$			

Here we say that $\epsilon_A(1)$ and $Lg_1(2)$ are undefined. Note that initially there will always be some undefined entries, and that these will all be in the ϵ-tables.

4.2.3. COINCIDENCES

Coincidences consist of two *different* elements which are to be made equal, and may occur in either of two ways. First, they may appear in Tables (iii) or (iv), when two elements appear in the same row of a table, one at the end of the left-hand side, the other at the end of the right-hand end. For example:

$LB_1 \xrightarrow{Lg_1} \cdots \xrightarrow{Lg_n} LB_2$			$LB_1 \xrightarrow{Lh_1} \cdots \xrightarrow{Lh_m} LB_2$		
1		3	1		3
2	$\cdots$	<u>4</u>	2	$\cdots$	<u>6</u>
3			3		

Here we have a coincidence, namely 4 and 6. We denote such a coincidence by 4 — 6. The second way coincidences may occur is in "dealing with coincidences" (discussed in more detail below). In this case they arise in the L-tables when two different elements occur in the same column of two rows which are already coincidences themselves. For example, if we have a coincidence of 1 and 2, and have the following table:

	Lg_1	Lg_2		Lg_n
LB	LB_1	LB_2	$\cdots$	LB_n
1	3	$\underline{4}$		7
2	3	$\underline{5}$		7
$\vdots$				

then there is the further coincidence of 4 and 5.

4.3. THE PROCEDURE

In Theorem 2.1, the left Kan extension was described on objects as the quotient of

$$\sum_{A\in\mathbf{A}} \mathbf{B}(FA, B) \times XA$$

by an equivalence relation. The generalized Todd–Coxeter procedure essentially enumerates the elements of the closely related set

$$\sum_{A\in\mathbf{A}} \mathcal{F}G_B(FA, B) \times XA$$

and periodically employs a sub-procedure to impose both the equivalence relation of Theorem 2.1 and the relations between generating arrows of $\mathbf{B}$.

The general procedure is given by:

```
initialize tables
while there are undefined elements
    (
    define new elements
    fill in consequences
    while there are coincidences
        (
        deal with coincidences
        fill in consequences
        )
    )
```

Above we described how to *initialize tables* and the terms *undefined elements* and *coincidences* were explained. It remains to describe the procedures *filling in consequences, defining new elements* and *dealing with coincidences.*

4.3.1. FILLING IN CONSEQUENCES

It has been noted that the tables of type (i) and (ii) represent partial functions between the various sets XA and LB. When we *fill in consequences* we use these partial functions to fill all possible entries in the type (iii) and (iv) tables.

4.3.2. DEFINING NEW ELEMENTS

Given a undefined element in a column headed by, say, LB_1, we define a new element, adding to LB_1 the first integer not already in LB_1 and placing this in the unfilled position, in the left-hand column of the LB_1-table and in the left-hand columns of each side of each type (iii) and (iv) table which begin with LB_1. We clarify this with an example: if we have tables

	Lg_1	Lg_2		Lg_n
LB	LB_1	LB_2	$\cdots$	LB_n
1	2	4	$\cdots$	7
2	$\square$			

Lh	
$LB_1 \to LB_2$	
1	3
2	

and

Lf_1		Lf_2	
$LB_1 \to \cdots$		$LB_1 \to \cdots$	
1		1	
2		2	

Then we define a new element for $Lg_1(2)$ and these tables become:

	Lg_1	Lg_2		Lg_n
LB	LB_1	LB_2	$\cdots$	LB_n
1	2	4	$\cdots$	7
2	3			

Lh	
$LB_1 \to LB_2$	
1	3
2	
3	

and

Lf_1		Lf_2	
$LB_1 \to \cdots$		$LB_1 \to \cdots$	
1		1	
2		2	
3		3	

It is also essential to clarify which new elements to define and when to define them. In this step of the procedure, we may define any number of new elements (though at least one must be defined), and the only necessary guiding principle in choosing which ones to define is that for each $x \in XA$ and $g = g_n \cdots g_1$ in $\mathcal{F}G_B$ the element $Lg_n \cdots Lg_1\epsilon_A(x)$ would be defined after a finite number of steps of the procedure (if it does not terminate first of course). A typical systematic method is to first define elements in the ϵ-tables, and then to work row by row across the L-tables.

4.3.3. DEALING WITH COINCIDENCES

We take a coincidence, say $x - y$, where $x, y \in LB$, from the list of current coincidences. We then add to the list of any coincidences arising from the L-tables as a result of the $x - y$ coincidence in the manner described above. Next, replace every occurrence of, say, y as an element of LB in the tables and the coincidence list (traditionally the smaller of the two replaces the larger, but this is not essential). Finally, for appearances sake, we generally relabel the elements of LB so as to obtain a set of consecutive integers once more.

5. Calculation examples

In this section we return to the examples of left Kan extensions given in Section 3 and in each case work through an application of the generalized procedure to clarify the processes involved. We are restricted to considering very simple examples, as the length of working involved for any significant examples becomes prohibitive in the context of a written explanation. Such examples are best suited to computer implementation. †

5.1. MONOIDS

Recall from Section 3 (i) that with the following left Kan extension:

$$\begin{tikzcd} \mathbf{1} \arrow[rr, "\subseteq"] \arrow[dr, "X"'] & & \mathbf{M} \arrow[dl, "L"] \\ & \mathbf{Sets} & \end{tikzcd}$$

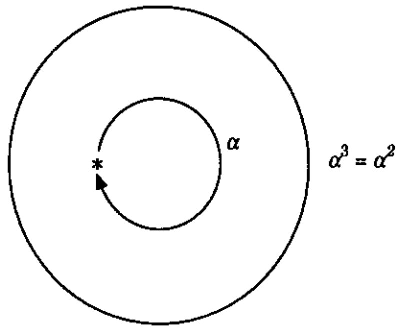
where M is a monoid and $X\bullet$ is a one-point set, $L*$ simply gives the elements of the monoid. The generalized Todd–Coxeter procedure needs only generators and relations for M, the procedure thus may be used to enumerate the elements of a monoid given generators and relations.

We will illustrate this application with a simple example. Consider the monoid given by

$$M = \langle \alpha \mid \alpha^3 = \alpha^2 \rangle;$$

that is, we have the category $\mathbf{M}$ specified as:

† Indeed, Sean Carmody and Craig Reilly wrote a prototype computer program for the general case which successfully calculated a number of longer examples. Other programs dealing with special cases were also written by Lesley Johnston and Diana Gibson.



Since the category **1** has no generating arrows, there will be no tables of type (iv). We thus begin tables initialized as follows:

(i)

$X\bullet \xrightarrow{\epsilon_\bullet} L*$	
1	

(ii)

$L* \xrightarrow{L\alpha} L*$	

(iii)

$L* \xrightarrow{L\alpha} L* \xrightarrow{L\alpha} L*$		

$L* \xrightarrow{L\alpha} L* \xrightarrow{L\alpha} L* \xrightarrow{L\alpha} L*$			

We first begin defining new elements and filling in consequences until the first coincidence is met.

(i)

$X\bullet \xrightarrow{\epsilon_\bullet} L*$	
1	1

(ii)

$L* \xrightarrow{L\alpha} L*$	
1	2
2	3
3	4
4	

(iii)

$L* \xrightarrow{L\alpha} L* \xrightarrow{L\alpha} L*$		
1	2	$\underline{3}$

$L* \xrightarrow{L\alpha} L* \xrightarrow{L\alpha} L* \xrightarrow{L\alpha} L*$			
1	2	3	$\underline{4}$

Dealing with the coincidence 3 — 4 is simple, producing no further coincidences:

(i)

$X\bullet \xrightarrow{\epsilon_\bullet} L*$	
1	1

(ii)

$L* \xrightarrow{L\alpha} L*$	
1	2
2	3
3	3

(iii)

$L* \xrightarrow{L\alpha} L* \xrightarrow{L\alpha} L*$		
1	2	3

$L* \xrightarrow{L\alpha} L* \xrightarrow{L\alpha} L* \xrightarrow{L\alpha} L*$			
1	2	3	3

There are now no undefined elements and no coincidences to deal with, so the process is complete, and the conclusion is that M has three elements, the element "1" corresponding to the identity of M. The completed L-table describes how to multiply the elements with α. In this case, we have simply discovered that

$$M = \{1, \alpha, \alpha^2\}.$$

5.2. CATEGORIES

We now consider the left Kan extension of Section 3 (ii):

$$\begin{tikzcd}
\mathrm{obj}\,\mathbf{C} \arrow[rr, "\subseteq"] \arrow[dr, "X"'] & & \mathbf{C} \arrow[dl, "L"] \\
& \mathbf{Sets} &
\end{tikzcd}$$

where X takes objects to one-point sets. Here LA consists of all the arrows out of A, for each object A of $\mathbf{C}$. Thus the generalized procedure may be used to enumerate the arrows of a category given by generators and relations.

We again consider a simple example. We specify the category $\mathbf{C}$ as:

$$\begin{tikzcd}
& A \arrow[dr, "\alpha"] & \\
C \arrow[ur, "\gamma"] & & B \arrow[ll, "\beta"]
\end{tikzcd}$$

$$\gamma\beta\alpha = 1_A$$
$$\alpha\gamma\beta = 1_B$$
$$\beta\alpha\gamma = 1_C$$

Once again there are no tables of type (iv). In the tables which follow, notice the way the treatment of identities in relations essentially as empty words in the generators. So we have initially:

(i)

ϵ_A

$XA \to LA$	
1	

ϵ_B

$XB \to LB$	
1	

ϵ_C

$XC \to LC$	
1	

(ii)

$L\alpha$

$LA \to LB$	

$L\beta$

$LB \to LC$	

$L\gamma$

$LC \to LA$	

(iii)

$LA \xrightarrow{L\alpha} LB \xrightarrow{L\beta} LC \xrightarrow{L\gamma} LA$				LA

$LB \xrightarrow{L\beta} LC \xrightarrow{L\gamma} LA \xrightarrow{L\alpha} LB$				LB

$LC \xrightarrow{L\gamma} LA \xrightarrow{L\alpha} LB \xrightarrow{L\beta} LC$				LC

We begin defining elements until we obtain:

(i)

$XA \xrightarrow{\epsilon_A} LA$	
1	1

$XB \xrightarrow{\epsilon_B} LB$	
1	1

$XC \xrightarrow{\epsilon_C} LC$	
1	1

(ii)

$LA \xrightarrow{L\alpha} LB$	
1	2
2	3
3	4
4	

$LB \xrightarrow{L\beta} LC$	
1	2
2	3
3	4
4	

$LC \xrightarrow{L\gamma} LA$	
1	2
2	3
3	4
4	

(iii)

$LA \xrightarrow{L\alpha} LB \xrightarrow{L\beta} LC \xrightarrow{L\gamma} LA$				LA
1	2	3	$\underline{4}$	$\underline{1}$
2	3	4		2
3	4			3
4				4

$LB \xrightarrow{L\beta} LC \xrightarrow{L\gamma} LA \xrightarrow{L\alpha} LB$				LB
1	2	3	$\underline{4}$	$\underline{1}$
2	3	4		2
3	4			3
4				4

$LC \xrightarrow{L\gamma} LA \xrightarrow{L\alpha} LB \xrightarrow{L\beta} LC$				LC
1	2	3	$\underline{4}$	$\underline{1}$
2	3	4		2
3	4			3
4				4

This results in coincidences $1 - 4$ in each of LA, LB and LC. We now deal with these coincidences, which produce no further coincidences.

(i)

ϵ_A

XA	$\to LA$
1	1

ϵ_B

XB	$\to LB$
1	1

ϵ_C

XC	$\to LC$
1	1

(ii)

$L\alpha$

LA	$\to LB$
1	2
2	3
3	1

$L\beta$

LB	$\to LC$
1	2
2	3
3	1

$L\gamma$

LC	$\to LA$
1	2
2	3
3	1

(iii)

$L\alpha \quad L\beta \quad L\gamma$

LA	$\to LB$	$\to LC$	$\to LA$	LA
1	2	3	1	1
2	3	1	2	2
3	1	2	3	3

$L\beta \quad L\gamma \quad L\alpha$

LB	$\to LC$	$\to LA$	$\to LB$	LB
1	2	3	1	1
2	3	1	2	2
3	1	2	3	3

$L\gamma \quad L\alpha \quad L\beta$

LC	$\to LA$	$\to LB$	$\to LC$	LC
1	2	3	1	1
2	3	1	2	2
3	1	2	3	3

With no undefined elements and no coincidences, the process is now complete. Thus we have enumerated the arrows of the category **C** and have tables describing composition with the generators. For example, LA gives all the arrows into A. By observing the positions that they appear in the composition tables, we may identify 1, 2, 3 of LA with $1_A, \gamma, \gamma\beta$.

5.3. COSET ENUMERATION

Consider the example in Section 3 (iii):

$$\begin{tikzcd} \mathbf{H} \arrow[rr, "\subseteq"] \arrow[dr, "X"'] & & \mathbf{G} \arrow[dl, "L"] \\ & \mathbf{Sets} & \end{tikzcd}$$

where $H \leq G$ and $X*$ is a one-point set. In this case $L*$ gives the cosets G/H. The procedure may thus be used to enumerate these cosets given generators and relations for G and generators of H in terms of those of G. Here the procedure is precisely the traditional Todd–Coxeter procedure, differing only slightly in the presentation of the tables.

In this case, we consider an example which is slightly more substantial than the previous two. Consider the dihedral group $G = D_8$ given as:

$$G = \langle \sigma, \tau \mid \tau^2 = \sigma^4 = 1, \tau\sigma\tau = \sigma^3 \rangle$$

and the subgroup

$$H = \langle \tau\sigma \rangle.$$

We specify this input by considering $H = \langle h \rangle$ and setting $Fh = \sigma\tau$. Then the tables become:

(i)

$X* \xrightarrow{\epsilon_*} L*$	
1	

(ii)

$L*$	$\overset{L\sigma}{L*}$	$\overset{L\tau}{L*}$

(iii)

$L* \xrightarrow{L\tau} L* \xrightarrow{L\tau} L*$			$L*$

$L* \xrightarrow{L\sigma} L* \xrightarrow{L\sigma} L* \xrightarrow{L\sigma} L* \xrightarrow{L\sigma} L*$					$L*$

$L* \xrightarrow{L\tau} L* \xrightarrow{L\sigma} L* \xrightarrow{L\tau} L*$				$L* \xrightarrow{L\sigma} L* \xrightarrow{L\sigma} L* \xrightarrow{L\sigma} L*$			

(iv)

$X* \xrightarrow{Xh} X* \xrightarrow{\epsilon_*} L*$			$X* \xrightarrow{\epsilon_*} L* \xrightarrow{L\sigma} L* \xrightarrow{L\tau} L*$			
1	1		1			

We begin defining new elements:

(i)

$X* \xrightarrow{\epsilon_*} L*$	
1	1

(ii)

$L*$	$\overset{L\sigma}{L*}$	$\overset{L\tau}{L*}$
1	2	3
2	4	5
3		
4		
5		

(iii)

$L* \xrightarrow{L\tau} L* \xrightarrow{L\tau} L*$			$L*$
1	3		1
2	5		2
3			3
4			4
5			5

$L* \xrightarrow{L\sigma} L* \xrightarrow{L\sigma} L* \xrightarrow{L\sigma} L* \xrightarrow{L\sigma} L*$					$L*$
1	2	4			1
2	4				2
3					3
4					4
5					5

$L* \xrightarrow{L\tau} L* \xrightarrow{L\sigma} L* \xrightarrow{L\tau} L*$				$L* \xrightarrow{L\sigma} L* \xrightarrow{L\sigma} L* \xrightarrow{L\sigma} L*$			
1	3			1	2	4	
2	5			2	4		
3				3			
4				4			
5				5			

(iv)

$X* \xrightarrow{Xh} X* \xrightarrow{\epsilon_*} L*$			$X* \xrightarrow{\epsilon_*} L* \xrightarrow{L\sigma} L* \xrightarrow{L\tau} L*$			
1	1	$\underline{1}$	1	1	2	$\underline{5}$

This results in the coincidence 1 — 5, which we now deal with:

(i)

$X* \xrightarrow{\epsilon_*} L*$	
1	1

(ii)

$L*$	$\overset{L\sigma}{L*}$	$\overset{L\tau}{L*}$
1	2	3
2	4	1
3		
4		

(iii)

$L* \xrightarrow{L\tau} L* \xrightarrow{L\tau} L*$			$L*$
1	3		1
2	1	$\underline{3}$	$\underline{2}$
3			3
4			4

$L* \xrightarrow{L\sigma} L* \xrightarrow{L\sigma} L* \xrightarrow{L\sigma} L* \xrightarrow{L\sigma} L*$					$L*$
1	2	4			1
2	4				2
3					3
4					4

$L*$	$\xrightarrow{L\tau} L*$	$\xrightarrow{L\sigma} L*$	$\xrightarrow{L\tau} L*$	$L*$	$\xrightarrow{L\sigma} L*$	$\xrightarrow{L\sigma} L*$	$\xrightarrow{L\sigma} L*$
1	3			1	2	4	
2	1	2	4	2	4		
3				3			
4				4			

(iv)

$X*$	$\xrightarrow{Xh} X*$	$\xrightarrow{\epsilon_*} L*$	$X*$	$\xrightarrow{\epsilon_*} L*$	$\xrightarrow{L\sigma} L*$	$\xrightarrow{L\tau} L*$
1	1	1	1	1	2	1

There is now a coincidence of 2 — 3; dealing with this gives:

(i)

$X*$	$\xrightarrow{\epsilon_*} L*$
1	1

(ii)

$L*$	$L\sigma$ $L*$	$L\tau$ $L*$
1	2	2
2	3	1
3		

(iii)

$L*$	$\xrightarrow{L\tau} L*$	$\xrightarrow{L\tau} L*$	$L*$
1	2	1	1
2	1	2	2
3			3

$L*$	$\xrightarrow{L\sigma} L*$	$\xrightarrow{L\sigma} L*$	$\xrightarrow{L\sigma} L*$	$\xrightarrow{L\sigma} L*$	$L*$
1	2	3			1
2	3				2
3					3

$L*$	$\xrightarrow{L\tau} L*$	$\xrightarrow{L\sigma} L*$	$\xrightarrow{L\tau} L*$	$L*$	$\xrightarrow{L\sigma} L*$	$\xrightarrow{L\sigma} L*$	$\xrightarrow{L\sigma} L*$
1	2	3		1	2	3	
2	1	2	1	2	3		
3				3			

(iv)

$X*$	$\xrightarrow{Xh} X*$	$\xrightarrow{\epsilon_*} L*$	$X*$	$\xrightarrow{\epsilon_*} L*$	$\xrightarrow{L\sigma} L*$	$\xrightarrow{L\tau} L*$
1	1	1	1	1	2	1

We define a new element for $L\tau(3)$, and looking ahead a little to save space, it is clear from the first line of the $\tau\sigma\tau = \sigma^3$ table that $L\tau(3) = L\sigma$. We then have:

(i)

$X*$	$\xrightarrow{\epsilon_*} L*$
1	1

(ii)

$L*$	$L*$ ($L\sigma$)	$L*$ ($L\tau$)
1	2	2
2	3	1
3	4	4
4		

(iii)

$L*$	$\xrightarrow{L\tau} L*$	$\xrightarrow{L\tau} L*$	$L*$
1	2	1	1
2	1	2	2
3	4		3
4			

$L*$	$\xrightarrow{L\sigma} L*$	$\xrightarrow{L\sigma} L*$	$\xrightarrow{L\sigma} L*$	$\xrightarrow{L\sigma} L*$	$L*$
1	2	3	4		1
2	3	4			2
3	4				3
3	4				3

$L*$	$\xrightarrow{L\tau} L*$	$\xrightarrow{L\sigma} L*$	$\xrightarrow{L\tau} L*$
1	2	3	4
2	1	2	1
3	4		
4			

$L*$	$\xrightarrow{L\sigma} L*$	$\xrightarrow{L\sigma} L*$	$\xrightarrow{L\sigma} L*$
1	2	3	4
2	3	4	
3	4		
4			

(iv)

$X*$	$\xrightarrow{Xh} X*$	$\xrightarrow{\epsilon_*} L*$
1	1	1

$X*$	$\xrightarrow{\epsilon_*} L*$	$\xrightarrow{L\sigma} L*$	$\xrightarrow{L\tau} L*$
1	1	2	1

Skipping a few steps, it is clear from the first row of th $\sigma^4 = 1$ table that $L\sigma(4) = 1$ and from the third row of the $\tau^2 = 1$ table that $L\tau(3) = 3$. Filling in the consequences yields:

(i)

$X*$	$\xrightarrow{\epsilon_*} L*$
1	1

(ii)

$L*$	$L*$ ($L\sigma$)	$L*$ ($L\tau$)
1	2	2
2	3	1
3	4	4
4	1	3

(iii)

$L*$	$\xrightarrow{L\tau} L*$	$\xrightarrow{L\tau} L*$	$L*$
1	2	1	1
2	1	2	2
3	4	3	3
4	3	3	3

$L*$	$\xrightarrow{L\sigma} L*$	$\xrightarrow{L\sigma} L*$	$\xrightarrow{L\sigma} L*$	$\xrightarrow{L\sigma} L*$	$L*$
1	2	3	4	1	1
2	3	4	1	2	2
3	4	1	2	3	3
4	1	2	3	4	4

$L*$	$\xrightarrow{L\tau} L*$	$\xrightarrow{L\sigma} L*$	$\xrightarrow{L\tau} L*$	$L*$	$\xrightarrow{L\sigma} L*$	$\xrightarrow{L\sigma} L*$	$\xrightarrow{L\sigma} L*$
1	2	3	4	1	2	3	4
2	1	2	1	2	3	4	1
3	4	1	2	3	4	1	2
4	3	4	4	4	1	2	3

(iv)

$X*$	$\xrightarrow{Xh} X*$	$\xrightarrow{\epsilon_*} L*$	$X*$	$\xrightarrow{\epsilon_*} L*$	$\xrightarrow{L\sigma} L*$	$\xrightarrow{L\tau} L*$
1	1	1	1	1	2	1

The procedure has now terminated, indicating that G/H consists of four cosets, which from table (ii) must be $\{H, \sigma H, \sigma^2 H, \sigma^3 H\}$. This is as expected, since $\tau\sigma$ in fact has order two.

5.4. GROUP ACTIONS

We now consider example (iv) of Section 3:

$$\begin{tikzcd} \mathbf{G} \arrow[rr, "!"] \arrow[dr, "X"'] & & \mathbf{1} \arrow[dl, "L"] \\ & \mathbf{Sets} & \end{tikzcd}$$

where X describes an action of G on $\Omega = X*$. Here L calculates the orbits Ω/G. Thus the generalized Todd–Coxeter procedure may be used to calculate orbits of an action which is specified by giving the set Ω and the action of the generators of G on Ω.

As an example, we consider the group $G = S_3$, the symmetric group on three elements, which we specify as generated by τ (a two cycle) and σ (a three cycle). We take an action on $\Omega = [4]$ given by $X\tau = (1,2)$ and $X\sigma = (1,2,3)$ using cycle permutation notation. Notice that in accordance with the input specifications for the procedure, no relations between the generators of G are required, and it is quite clear here that this is because all the necessary information is contained in the functor X. In this case, since $\mathbf{1}$ has no generating arrows, we have no tables of type (iii):

(i)

$X*$	$\xrightarrow{\epsilon_*} L\bullet$
1	
2	
3	
4	

(ii)

$L\bullet$

(iv)

$X\tau \quad \epsilon_* \qquad\qquad \epsilon_*$

$X* \rightarrow$	$X* \rightarrow$	$L\bullet$	$X* \rightarrow$	$L\bullet$
1	2		1	
2	1		2	
3	3		3	
4	4		4	

$X\sigma \quad \epsilon_* \qquad\qquad \epsilon_*$

$X* \rightarrow$	$X* \rightarrow$	$L\bullet$	$X* \rightarrow$	$L\bullet$
1	2		1	
2	3		2	
3	1		3	
4	4		4	

We now fill the tables:

(i)

ϵ_*

$X* \rightarrow$	$L\bullet$
1	1
2	2
3	3
4	4

(ii)

$L\bullet$
1
2
3
4

(iv)

$X\tau \quad \epsilon_* \qquad\qquad \epsilon_*$

$X* \rightarrow$	$X* \rightarrow$	$L\bullet$	$X* \rightarrow$	$L\bullet$
1	2	$\underline{2}$	1	$\underline{1}$
2	1	$\underline{1}$	2	$\underline{2}$
3	3	$\underline{3}$	3	$\underline{3}$
4	4	4	4	4

$X\sigma \quad \epsilon_* \qquad\qquad \epsilon_*$

$X* \rightarrow$	$X* \rightarrow$	$L\bullet$	$X* \rightarrow$	$L\bullet$
1	2	$\underline{2}$	1	$\underline{1}$
2	3	$\underline{3}$	2	$\underline{2}$
3	1	$\underline{1}$	3	$\underline{3}$
4	4	4	4	4

This results in the coincidences 1 — 2, 2 — 3 and 1 — 3. Dealing with these results in:

(i)

ϵ_*	
$X* \to L\bullet$	
1	1
2	1
3	1
4	2

(ii)

$L\bullet$
1
2

(iv)

$X\tau$	ϵ_*		ϵ_*	
$X* \to X* \to L\bullet$			$X* \to L\bullet$	
1	2	1	1	1
2	1	1	2	1
3	3	1	3	1
4	4	2	4	2

$X\sigma$	ϵ_*		ϵ_*	
$X* \to X* \to L\bullet$			$X* \to L\bullet$	
1	2	1	1	1
2	3	1	2	1
3	1	1	3	1
4	4	2	4	2

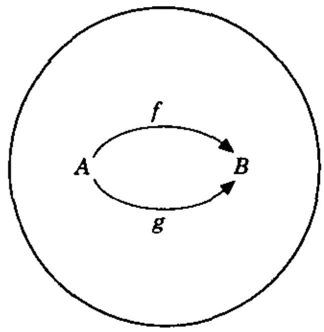
The procedure is now complete and we conclude that Ω has two orbits, one is $\{1, 2, 3\}$ and the other $\{4\}$.

5.5. COEQUALIZERS IN SET

The final example we consider is from Section 3 (v):

$$\begin{tikzcd} \bullet \arrow[r, shift left] \arrow[r, shift right] & \bullet \arrow[rr, "!"] \arrow[dr, "X"'] & & \mathbf{1} \arrow[dl, "L"] \\ & & \mathbf{Sets} & \end{tikzcd}$$

where $\bullet \rightrightarrows \bullet$ represents the category



In our example, we take

$$\begin{aligned} XA &= [3] \\ XB &= [4] \end{aligned}$$

and

$$Xf \quad : XA \to XB : \begin{cases} 1 \mapsto 1 \\ 2 \mapsto 2 \\ 3 \mapsto 3 \end{cases}$$

$$Xg \quad : XA \to XB : \begin{cases} 1 \mapsto 1 \\ 2 \mapsto 1 \\ 3 \mapsto 3. \end{cases}$$

Initializing the tables gives:

(i)

ϵ_A

$XA \to L\bullet$	
1	
2	
3	

ϵ_B

$XB \to L\bullet$	
1	
2	
3	
4	

(ii)

$L\bullet$

(iv)

Xf	ϵ_B		ϵ_A	
$XA \to$	$XB \to$	$L\bullet$	$XA \to$	$L\bullet$
1	1		1	
2	2		2	
3	3		3	

Xg ϵ_B ϵ_A

$XA \to$	$XB \to$	$L\bullet$	$XA \to$	$L\bullet$
1	1		1	
2	1		2	
3	3		3	

We begin defining new elements as usual:

(i)

ϵ_A

$XA \to$	$L\bullet$
1	1
2	2
3	3

ϵ_B

$XB \to$	$L\bullet$
1	4
2	
3	
4	

(ii)

$L\bullet$
1
2
3
4

(iv)

Xf ϵ_B ϵ_A

$XA \to$	$XB \to$	$L\bullet$	$XA \to$	$L\bullet$
1	1	$\underline{4}$	1	$\underline{1}$
2	2		2	2
3	3		3	3

Xg ϵ_B ϵ_A

$XA \to$	$XB \to$	$L\bullet$	$XA \to$	$L\bullet$
1	1	$\underline{4}$	1	$\underline{1}$
2	1		2	2
3	3		3	3

This gives the coincidence 1 — 4, which we deal with:

(i)

ϵ_A

$XA \to$	$L\bullet$
1	1
2	2
3	3

ϵ_B

$XB \to$	$L\bullet$
1	1
2	
3	
4	

(ii)

$L\bullet$
1
2
3

(iv)

Xf ϵ_B ϵ_A

$XA \to$	$XB \to$	$L\bullet$	$XA \to$	$L\bullet$
1	1	1	1	1
2	2		2	2
3	3		3	3

Xg ϵ_B ϵ_A

$XA \to$	$XB \to$	$L\bullet$	$XA \to$	$L\bullet$
1	1	1	1	1
2	1	$\underline{1}$	2	$\underline{2}$
3	3		3	3

We now have the coincidence 1 — 2, which we also deal with:

(i)

ϵ_A

$XA \to L\bullet$	
1	1
2	1
3	2

ϵ_B

$XB \to L\bullet$	
1	1
2	
3	
4	

(ii)

$L\bullet$
1
2

(iv)

Xf ϵ_B ϵ_A

$XA \to XB \to L\bullet$			$XA \to L\bullet$	
1	1	1	1	1
2	2		2	1
3	3		3	2

Xg ϵ_B ϵ_A

$XA \to XB \to L\bullet$			$XA \to L\bullet$	
1	1	1	1	1
2	1	1	2	1
3	3		3	2

At this point it is clear that the final outcome will be:

(i)

ϵ_A

$XA \to L\bullet$	
1	1
2	1
3	2

ϵ_B

$XB \to L\bullet$	
1	1
2	1
3	2
4	3

(ii)

$L\bullet$
1
2
3

(iv)

Xf ϵ_B ϵ_A

$XA \to XB \to L\bullet$			$XA \to L\bullet$	
1	1	1	1	1
2	2	1	2	1
3	3	2	3	2

Xg ϵ_B ϵ_A

$XA \to XB \to L\bullet$			$XA \to L\bullet$	
1	1	1	1	1
2	1	1	2	1
3	3	2	3	2

The conclusion is that the coequalizer of Xf and Xg partitions XB into three subsets, shown by ϵ_B to be $\{1,2\}$, $\{3\}$ and $\{4\}$. Of course, 1 and 2 are in the same partition because $2 \in XA$ maps to both of them via Xg and Xf respectively.

6. Graphs and presentations

The behaviour of the general procedure is highly recursive and quite subtle, and its description in terms of filling entries in tables was not found to be particularly amenable to analysis. Accordingly, the authors developed a description in terms of *presentations of functors* or simply *presentations*. These presentations, which are fully defined in Carmody and Walters (1990), are naturally motivated by a graphical realization of the tables used in the procedure (see also Dimino, 1971, for a graphical approach to the classical Todd–Coxeter procedure), and it is at this level that we give a brief description of them here. Consider, for example, L-tables given by:

Lf

$LA \to LB$	
1	1
2	

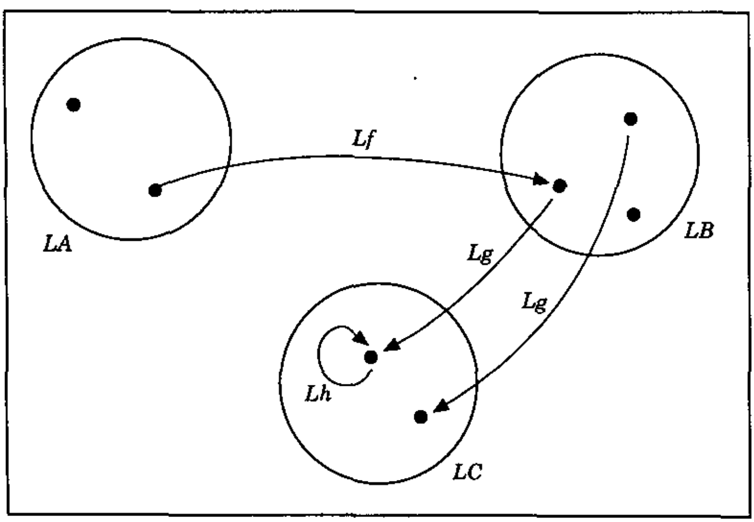
Lg

$LB \to LC$	
1	1
2	2
3	

Lh

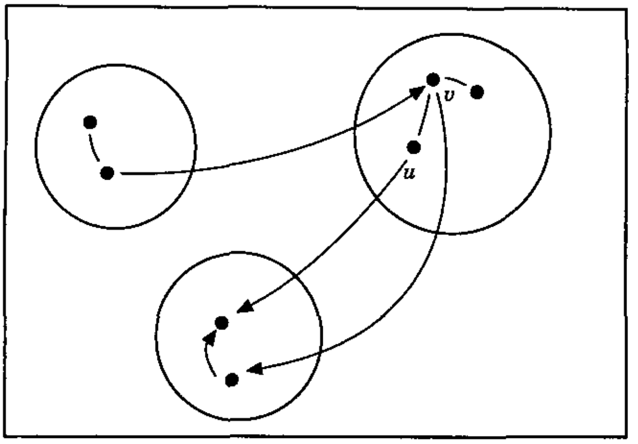
$LC \to LC$	
1	1
1	

These tables can be thought of as consisting of three sets LA, LB LC and partial functions between them. These can then be represented graphically as follows:

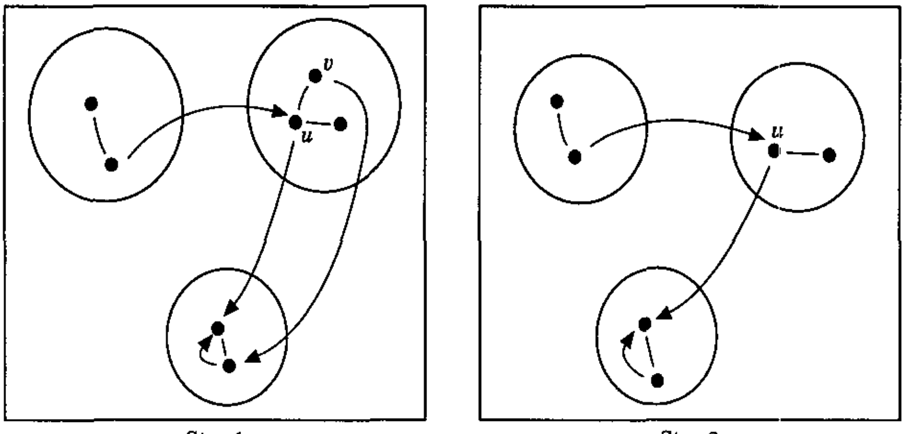


This diagram is an example of a *presentation of a functor*.

The process of *Dealing with Coincidences* can be described in terms of presentations. We represent coincidences on the diagrams by a line • — •. If we have the example



then we deal with the coincidence u — v with a sequence of two modifications. First, all arrows and lines into v are moved to u (keeping only the single line u — v) and "consequences" of u — v are added, then v itself is removed along with arrow and lines out of v. These two steps result in an "equivalent" presentation in which v has been removed:



Step 1 Step 2

In Carmody and Walters (1990), presentations are used to formally describe this subprocedure *Dealing with Coincidences* and also to prove its correctness and termination.

References

Buchberger, B. (1987). History and basic features of the critical-pair/completion procedure. *J. Symbolic Computation* **3**, 3–38.

Carmody, S., Walters, R. F. C. (1990). Computing quotients of actions on a free category. *Univ. Sydney, Dept. Pure Maths Research Reports* **90-20**.

Coxeter, H. S. M., Moser, W. O. J. (1957). *Generators and Relations for Discrete Groups*. Heidelberg: Springer-Verlag.

Dimino, L. A. (1971). A Graphical Approach to Coset Enumeration. *Sigsam Bull.* 19, 8–43.

Johnson, D. L. (1990). *Presentation of Groups*. Cambridge: Cambridge University Press.

Leech, J. (1984). Coset Enumeration. In: *Computational Group Theory, Lon. Math. Soc. Symposium on Comp. Group Theory* (Atkinson, M. D., ed.), pp. 3–18. London: Academic Press.

Leeming, M., Walters, R. F. C. (1995). Computing left Kan extensions using the Todd–Coxeter procedure. In: *Computational Algebra and Number Theory* (van der Poorten, A., Bosma, W., eds), pp. 53–73. Dordrecht: Kluwer.

MacLane, S. (1971). *Categories for the Working Mathematician*. New York: Springer-Verlag.

Mora, F. (*c.* 1978). *Computation of Finite Categories*. Preprint.

Mendelsohn, N. S. (1964). An algorithmic solution for a word problem in group theory. *Can. J. Mathematics* **16**, *509–516*.

Sims, C. (1994). *Computation With Finitely Presented Groups*. Cambridge: Cambridge University Press.

Appendix

Procedure
/* Input: $\mathcal{A}, \mathcal{B}, \mathcal{R}, \mathcal{X}, \mathcal{F}$; Output: L, ϵ
begin
    set the right-hand column of each ϵ-table equal to its left-hand column
    while tables are not complete **do**
        while $SA \neq \emptyset$ for some A **do**
            Take u—$v \in SA$; say $u < v$
            $SA := Delete(SA)$
            replace all instances of v in columns headed by LA by u
            delete repeated lines in the L-tables and $\mathcal{R}$-tables
            renumber
            derive consequences
        endwhile
        take object A such that max(entries in LA) is smallest
        set $i := max(\text{entries in } LA) + 1$
        insert i in all columns headed by LA
        create rows with left-hand column i in each L and $\mathcal{R}$-tables whose left-hand column is headed by LA
        derive consequences
    endwhile
end

Subprocedure *Delete*
/* Input: SA,u—v; Output SA'
begin
    while v—$w \in SA$ for some $w \neq u$ **do**
        $SA := SA \backslash \{v\text{—}w\} \cup \{u\text{—}w\}$
    endwhile
    for all L-tables $LA \to LA_i$
        if $g_i(u)$ is defined and $g_i(v)$ is defined and $g_i(u) \neq g_i(v)$
            then let $SA_i = SA_i \cup \{(g_i(u)\text{—}g_i(v))\}$
        endif
    $SA' = SA \backslash \{(u, v)\}$
end

Subprocedure *Derive consequences*
begin
    insert all entries into $\mathcal{R}$ and naturality tables derived from the partial functions in the ϵ and L-tables
    for each object A in the right-hand column of a $\mathcal{R}$ or naturality table
        for each row of that table take u and v as the entries in the right-hand columns
        if u and v are defined, and $u \neq v$,
            then $SA := SA \cup \{u - v\}$
        endif
end