Chapter 3

Formal Systems

1. Introduction

The theory of formal systems provides a general framework within which the concepts of axiomatization and deductive process can be studied and developed in a strictly mathematical way. In particular, the concepts 'theorem' and 'proof' become mathematical objects for which results, perhaps more properly described as meta-mathematical, can be derived in much the same way as in the fields of number theory or non-singular matrices. From these results follow the general properties that apply to all deductive mechanisms, which are important for the inference engines of expert systems; so it is no exaggeration to say that this abstract theory of formal systems lies at the heart of the theory of expert systems.

The idea of a formal system is important in fields other than mathematical logic. In epistemology for example it is recognized that the progress and formalization of any science are closely linked, the ideal state being a completely formal expression of its underlying theory. This is obviously the case for mathematics, but is true also for some branches of physics. In computer science and applications all the processes that occur, including numerical computation, are simply manipulations of formal symbols, so problems of a combinatorial nature and of the representational powers of formal systems are fundamental here. This appplies even more strongly in AI, where the need is to model—that is, to formalize—knowledge and behaviour. These are deep practical problems and therefore those who work on them should wherever possible take account of and use the results and tools provided by the mathematicians who have worked in this field for several decades.

This account is of course incomplete but it should help to fix ideas; it can be used in conjunction with the bibliography for this chapter.

2. Definitions

A *formal system* S consists of the following:

1. an *alphabet* Σ_S, either finite or, if infinite, denumerable; in the latter case an enumeration is given and fixed once and for all;

2. a set of *well-formed formulae* in Σ_S, which is a recursive subset F_S of the set of all finite strings formed from the elements of Σ_S; i.e.

   $F_S \subset \Sigma_S^*$
   F_S is recursive;
3. a set of *axioms* A_S, which is a recursive subset of F_S:
   $A_S \subset F_S$
   A_S is recursive;
4. a finite set of *rules of inference* R_S, which are decidable predicates defined over F_S. This set is written:
   $R_S = \{r_1, r_2, \ldots, r_n\}$
   The application of R_S is written $r_i(f_1, f_2, \ldots, f_j, g)$ or more usually, as in this book, $f_1, f_2, \ldots, f_j \vdash_{r_i} g$, meaning 'given the formulae $f_1, f_2, \ldots, f_j$ the formula g can be deduced by applying the rule r_i.'

Points to be noted in connexion with these definitions are:

1. the set of well-formed formulae may be defined by means of a production grammar; if this is the case, care must be taken to avoid confusing the rules of the grammar with the inference rules of the system;
2. the conditions of recursivity and decidability imply that:
   (a) there is a program P_1 that for input $f \in \Sigma_S^*$ states within a finite time whether or not $f \in F_S$;
   (b) there is a program P_2 that for input $f \in F_S$ states within a finite time whether or not $f \in A_S$;
   (c) for each rule r_i there is a program Q_i that for input $(f_1, f_2, \ldots, f_j, g)$ states within a finite time whether or not $f_1, f_2, \ldots, f_j \vdash_{r_i} g$.

Example

Consider the formal system defined as follows:

- $\Sigma_S = \{1, +, =\}$
- F_S = the set of all strings consisting of a finite, non-zero number of repetitions of the symbol '1,' followed by the single symbol ' + ,' followed by a finite, non-zero number of repetitions of the symbol '1,' followed by the single symbol ' = ,' followed by a finite non-zero number of repetitions of the symbol '1.' This can be written as
  $F_S = \{1^m + 1^n = 1^p\}$ where m, n and p are non-zero positive integers; an alternative notation used in language theory is:
  $F_S = (1^+ + 1^+ = 1^+)$
- $A_S = \{1 + 1 = 11\}$ (i.e. there is only a single axiom)
- $R_S = \{r_1, r_2\}$ where r_1: $1^m + 1^n = 1^p \vdash_{r_1} 1^{m+1} + 1^n = 1^{p+1}$

  r_2: $1^m + 1^n = 1^p \vdash_{r_2} 1^m + 1^{n+1} = 1^{p+1}$

  meaning that the formulae on the right can be deduced from those on the left by applying rules r_1 and r_2, respectively.

We say that any string of formulae $f_1, f_2, \ldots, f_p$ is a *deduction from* $h_1, h_2, \ldots, h_n$ if for all $i \in \{1, 2, \ldots, p\}$:

(a) f_i is an axiom, or
(b) f_i is one of the formulae $h_1, h_2, \ldots, h_n$, or
(c) f_i has been deduced by the application of a rule $r_k \in R_S$ to the formulae
$f_{i_0}, f_{i_1}, \ldots, f_{i_j}$ that precede f_i in the string; that is
$f_{i_0}, f_{i_1}, \ldots, f_{i_j} \vdash_{r_k} f_i,\ i_0 < i,\ i_1 < i, \ldots, i_j < i$

We express this also by saying that $f_1, f_2, \ldots, f_p$ is a *deduction from the hypothesis* $h_1, h_2, \ldots, h_n$ and write:

$$h_1, h_2, \ldots, h_n \vdash_S f_p$$

A *theorem of S* is any formula t that can be deduced from the empty set $\emptyset$; this is written:

$$\vdash_S t$$

The set of theorems of S is written T_S.

A deduction from $\emptyset$ is also called simply a *deduction*; it consists of a string of formulae $f_1, f_2, \ldots, f_p$ for which every element f_i either is an axiom or has been derived by the process (c) above.

Example

With rules r_1 and r_2 from the previous example the sequence of formulae

f_1: 1 + 11 = 11
f_2: 1 + 111 = 111 (applying r_2 to f_1)
f_3: 11 + 111 = 1111 (applying r_1 to f_2)

provide a deduction of the formula 11 + 111 = 1111 from 1 + 11 = 11, which latter is not an axiom.

On the other hand, the sequence:

f_1: 1 + 1 = 11 (axiom)
f_2: 1 + 11 = 111 (application of r_2 to f_1)
f_3: 1 + 111 = 1111 (application of r_2 to f_2)
f_4: 11 + 111 = 11111 (application of r_1 to f_3)

is a deduction of 11 + 111 = 11111 from $\emptyset$ and therefore this formula is a theorem of S. So we can write 11 + 111 = 11111 $\in T_S$.

Representation as a tree or graph is often a help in visualizing a deduction. For example, suppose we have this deduction for a certain formal system S:

f_1: axiom 1

f_2: axiom 2
f_3: application of rule r_1 to f_1 and f_2
f_4: application of rule r_2 to f_2
f_5: application of rule r_3 to f_2, f_3 and f_4

This can be represented by the tree shown in Fig. 3. Each extremity of the tree (often called a 'leaf') must be either an axiom of the system or an hypothesis.

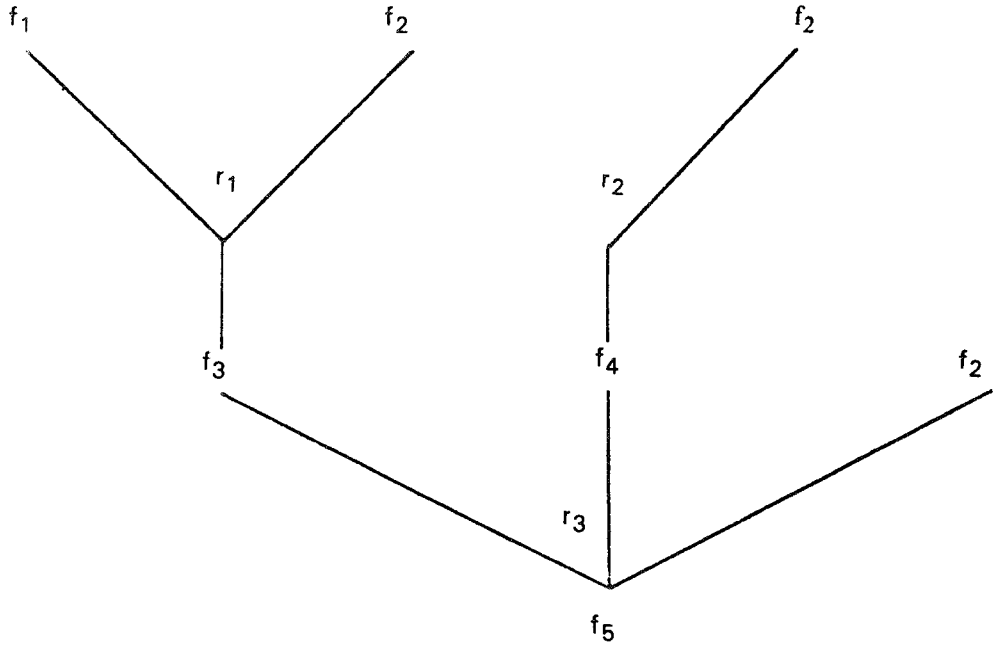

Fig. 3. Tree for the derivation of f_5.

The same deduction can be represented by a cycle-free graph having a number of nodes equal to the length of (i.e. number of steps in) the deduction, as shown in Fig. 4; but this form can become very tangled.

In the example we have given, as in all formal systems, the manipulations of the formulae are purely syntactic and take no account of any meaning that might be assigned to them. However, there is nothing against *interpreting* the formulae in any way that one chooses, and indeed it is because of this that formal systems are important. An interpretation can be useful as an aid to establishing a result but care must be taken not to misuse this possibility, and for example not to be misled by taking what one 'knows' to be true to be a formal theorem of the system. It can happen that the type of formalization, the axioms and the inference rules are inappropriate to the true nature of the system being represented; the degree of appropriateness may be only partial, as in the above example: it would be natural to expect 1 + 1 + 1 = 111 to be a theorem of the system S defined there, but this is not so for the simple reason that this is not a well-formed formula of S.

There are the following further definitions concerning a formal system S:

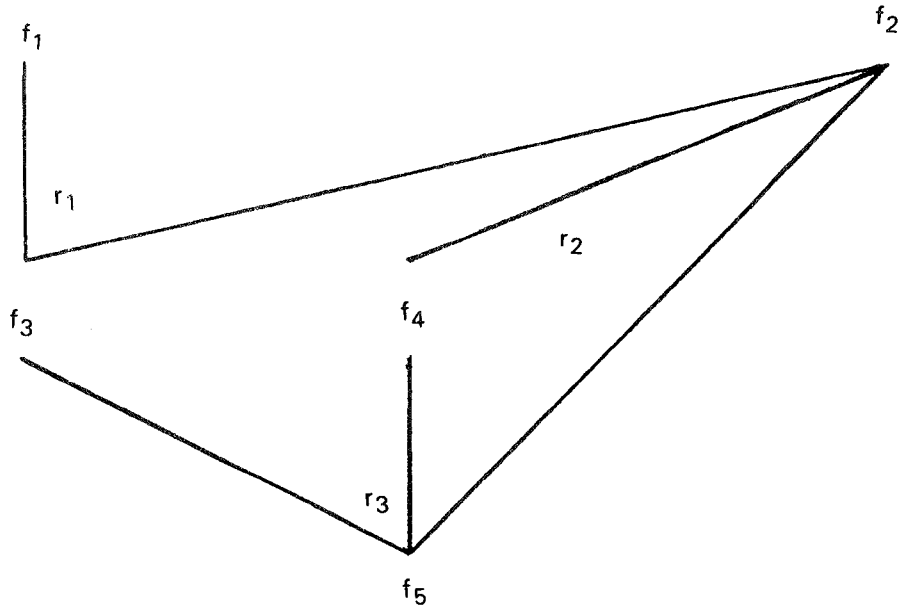

Fig. 4. Graph for the derivation of f_5.

1. The system S is *coherent* if it has well-formed formulae that are not theorems.
2. A system that includes a symbol for negation (usually written ' ¬ ' and pronounced 'not') is *consistent*, or *non-contradictory*, if there is no well-formed formula $f \in F_S$ which is such that both f and ¬ f are theorems of S. If this is not the case the system is said to be *inconsistent* or *self-contradictory* (so consistency implies coherence).
3. A system S that includes a symbol for negation is *categorical* if for every well-formed formula $f \in F_S$ (without free variables in the case of first-order predicate calculus) either $f \in T_S$ or ¬ $f \in T_S$.
4. A system S is *saturated* if for all well-formed formula $f \in T_S$ the system S' obtained by adding f to the axioms of S is inconsistent.
5. The axioms of a system S are *independent* if the removal of any axiom results in a formal system having fewer theorems than S.
6. The *decision problem* for S is the problem of finding whether or not T_S is recursive; put otherwise, of finding whether or not the predicate

   P(t): t is a theorem of S

   is decidable.

   If T_S is recursive, the system is said to be *decidable*.
7. A system S is *finitely axiomatizable* if A_S is finite or if there is a system having the same alphabet, well-formed formulae, inference rules and theorems that has a finite set of axioms.

A possible situation is that one wants to model a particular problem and would like a known set of formulae, T say, to be theorems of the formal

system used, S say. S is said to be *sound* for this purpose if $T_S \subset T$, and *complete* if $T_S = T$. These concepts and terms are used in Chapter 7.

From a purely syntactic point of view the two most important problems are those of consistency and decision. The concept of a formal system was introduced into mathematical logic before it was used in computer science, and a number of rather disturbing results were established. For example, Gödel in 1931 showed that the consistency of the formal system that constitutes arithmetic cannot be established without using methods more powerful than those of arithmetic itself; in particular, therefore, that the consistency of arithmetic cannot be established by any chain of reasoning that uses only the deductive rules and axioms of arithmetic. Gödel's proof showed that the goal set by Hilbert and the formalistic school of establishing the consistency of mathematics by purely finite methods could never be achieved.

From a more general point of view another problem of fundamental importance for formal systems is that of their appropriateness to the particular field in which they are being used for modelling. Here again some unexpected results have been proved. In his 1931 paper again Gödel showed, for example, that no formal system modelling arithmetic can be categorical; and further that if such a system is consistent and has as its theorems only formulae that are true—which is a minimal criterion for appropriateness—then there will be formulae that are true but are not theorems: that is, true formulae that cannot be proved.

3. Some general results

Proposition 1

If $h_1, h_2, \ldots, h_k \vdash_S g_1$

$h_1, h_2, \ldots, h_k \vdash_S g_2$

.....

$h_1, h_2, \ldots, h_k \vdash_S g_j$

and $g_1, g_2, \ldots, g_j \vdash_S f$

then $h_1, h_2, \ldots, h_k \vdash_S f$

In particular, if $h \vdash_S g$ and $g \vdash_S f$ then $h \vdash_S f$

Proof

For all $i \in \{1, 2, \ldots, j\}$ let $f_1^i, f_2^i, \ldots, f_{p_i}^i$ be a deduction of g_i from $h_1, h_2, \ldots, h_k$.

Also let $s_1, s_2, \ldots, s_q$ be a deduction of f, from $g_1, g_2, \ldots, g_j$. Then by using the definition of deduction it can be checked that:

$$f_1^1, f_2^1, \ldots, f_{p_1-1}^1, f_1^2, f_2^2, \ldots, f_{p_2-1}^2, f_1^j, f_2^j, \ldots, f_{p_j-1}^j, s_1, s_2, \ldots, s_q$$

is a deduction of f from $h_1, h_2, \ldots, h_k$. △

Proposition 2: the set of deductions is recursive

Proof

To prove this we have to show that there is a program that, given any finite string $f_1, f_2, \ldots, f_n$ where each f_i is an element of Σ_S^* will state within a finite time whether or not it is a deduction. Such a program can be constructed as follows; it uses the programs P_1, P_2, Q_1, ..., Q_n defined in connexion with the definition of a formal system (Section 1).

1. First check that each f_i is a well-formed formula (Program P_1).
2. Next find which if any of the f_i are axioms (Program P_2).
3. For each formula that is not an axiom check that it can be deduced from any collection of those that precede it in the sequence (Programs $Q_1, Q_2, \ldots Q_n$).

△

Proposition 3: The set T_S of theorems is recursively enumerable

Proof

This requires a proof that there is a program that will list all the theorems of S. Such a program can be constructed as follows:

1. Using the method described for constructing a program to list all the programs of $P_{(1,1)}$, suitably adapted, list all the finite strings of well-formed formulae one at a time.
2. Use each of these strings (as they are generated) as input to the program defined in the proof of Proposition 2, to find whether or not it is a deduction. If it is, print the last formula in the string (which will then be a theorem); if not, continue with the next in the list.

△

Proposition 4: there are formal systems S for which T_S is not recursive

This is a result of fundamental importance, for it shows that although the (abstract) objects in the definition of a formal system are recursive, the set of theorems may not be recursive. Further, for most of the formal systems of any interest or importance (not an attribute of the entirely artificial system used for the proof) T_S is not in fact recursive. We have therefore the following result:

> while we can always find an algorithm—therefore we can say: write a program—that will decide whether or not a given string of well-formed formulae is a deduction in general, there is no algorithm that will decide whether or not a given well-formed formula is a theorem.

Proof

To prove Proposition 4 we construct a formal system as follows:

- Σ_S: the symbols of the programming language that is being used, augmented by '[,' '],' ' | ,' '@,' 'halted,' 'not-halted.'
- F_S: all formulae of either of the following forms:

  $[P_n$ @ halted]
  $[P_n$ @ $|^m$ @ not-halted]

  where, as in Chapter 2, P_n is the nth program in the series of programs with 1 integer input and 1 integer output; and $|^m$ means 'm repetitions of | ' for any $m \geqslant 0$.

- A_S: the set of well-formed formulae:

  $[P_n$ @ @ not-halted]

- R_S: $[r_1, r_2]$ where r_1: $[P_n$ @ $|^m$ @ not-halted] $\vdash_{r_1}$ $[P_n$ @ $|^{m+1}$ @ not halted] if P_n with input n has not halted at the end of the (m + 1)th unit of time

  r_2: $[P_n$ @ $|^m$ @ not-halted] $\vdash_{r_2}$ $[P_n$ @ halted] if P_n with input n reaches a 'stop' instruction at the (m + 1)th unit of time

  (we have assumed here that time is measured in discrete units).

This system has been specially designed so that:

$[P_n$ @ $|^m$ @ not-halted] $\in T_S$ iff P_n with input n has not stopped by time m

Further:

$[P_n$ @ halted] $\in T_S$ iff P_n with input n reaches a 'stop' instruction

It follows that if the set T_S of theorems is recursive, then so also is the set $\{n: P_n(n) \neq \perp\}$; in Chapter 2 this was shown to be false, therefore T_S cannot be recursive. Δ

4. An illustrative example

We consider again the example in Section 2; we show that the set of theorems of this system is:

$$T_S = \{1^m + 1^n = 1^{m+n} \mid n \in \mathbb{N} - \{0\}, m \in \mathbb{N} - \{0\}\}$$

Recalling that 1^m means 'm repetitions of the symbol 1' and that the single axiom of S is 1 + 1 = 11, we see that $1^m + 1^n = 1^{m+n}$ is a theorem of S if m + n = 2. Suppose it to be a theorem for all pairs (m, n) with $m + n < p$,

with p some positive integer and m, $n > 0$; if m', n' are such that $m' + n' = p$ and $m' > 0$, $n' > 0$.

If $m' > 1$, then:

$1^{m'-1} + 1^{n'} = 1^{p-1}$ is a theorem because $(m' - 1) + n' = p - 1 < p$

Application of rule r_1 then gives:

$1^{m'-1+1} + 1^{n'} = 1^{p-1+1}$, i.e. $1^{m'} + 1^{n'} = 1^{p}$

must also prove for the case, $m' = 1$, $n' > 1$ using r_2 and $m' = 1$, $n' = 1$ using axiom so the formula is a theorem for $m' + n' = p$, and it follows by induction that $1^m + 1^n = 1^{m+n}$ is a theorem of S for all integers m, $n > 0$.

Now all the intermediate formulae in a deductive chain are of the form $1^m + 1^n = 1^{m+n}$ because:

1. If it is true for f it is true for g deduced from f by the application of either r_1 or r_2, as is obvious from the definition of these rules.
2. It is true for the axiom.

Thus the necessary and sufficient condition for a formula to be a theorem is that it is of the form $1^m + 1^n = 1^{m+n}$, m, $n > 0$, which is the required result.

Having succeeded in characterizing all the theorems of system S we can say:

1. S is coherent: for example, $1 + 1 = 111$ is a well-formed formula but not a theorem.
2. S is finitely axiomatizable: A_S is finite, consisting of a single axiom.
3. T_S is recursive because the decision problem can always be solved: given any $f \in \Sigma_S^*$ there is no difficulty in writing a program to decide within a finite time whether or not f is of the form $1^m + 1^n = 1^{m+n}$.
4. S is a system that correctly models the addition of non-zero positive integers; relative to this it is sound and complete.

With more complex systems, however, it is usually impossible to characterize the theorems.

Exercises

Interpretation of an Inference Engine as a Formal System

Taking the example in Chapter 1 define by each of the following three methods a formal system:

1. The axioms are the facts and the inference rules are the knowledge base.
2. The axioms are the facts and the knowledge base; there is a single inference rule, the rule of *modus ponens*.

3. The axioms are the facts and the knowledge base and the negation of the goal.

In each case give a deduction of H (implying three deductions).

A Simple Formal System

The system S is defined by:

$\Sigma_S = \{a, b, c\}$
$F_S = \{a^n\ b\ c^m \mid n, m \geq 0\}$
$A_S = \{a^{2i}\ b\ c^{2i}\}\ i \geq 0$
$R_S = \{r_1 : a^n\ b\ c^m,\ a^{n'} b\ c^{m'} \vdash\!\!\!- a^{n+n'}\ b\ c^m\}$

1. Show that $a^6\ b\ c^2$ and $a^{10}\ b$ are theorems.
2. Give a general expression for the theorems and show that any theorem can be deduced in at most three steps.
3. Show that if any one axiom is removed, the set T of theorems is changed.
4. Define a system $S'(\Sigma', F', A', R')$ where $\Sigma' = \Sigma$ and $F' = F$, such that A′ consists of a single formula, R′ of two rules and $T' = T$.

Hofstadter's System MIU

The system S is defined by:
$\Sigma = \{M, I, U\}$
$F = \Sigma^*$ (i.e. any finite string of elements of Σ is a well-formed formula)
$A = \{MI\}$
$R = \{r_1, r_2, r_3, r_4\}$ where for any formulae f, g

$r_1 : f \vdash\!\!\!-_{r_1} fU$,

$r_2 : Mf \vdash\!\!\!-_{r_2} Mff$,

$r_3 : fIIIg \vdash\!\!\!-_{r_3} fUg$,

$r_4 : fUUg \vdash\!\!\!-_{r_4} fg$.

1. Show that $M\ U\ I\ U \in T_S$.
2. Show that if $t \in T_S$, then t begins with M.
3. Is $MU \in T_S$?
4. Characterize the theorems of S.

A Formal System with an Infinite Number of Axioms

The system S is defined by:

$\Sigma = \{ = , + , 1, 2, \ldots, n, \ldots\}$

$F = \{n_0 + n_1 + \ldots + n_p = m_0 + m_1 + \ldots + m_q \mid$ p, q integers ≥ 0, m_i, n_i integers > 0

$A = \{1 + 1 + (p \text{ times}) + 1 = p \mid$ p integer > 0

$R = \{r_1: f_1, f_2 \vdash\!\!\!- g\}$ where

$f_1: n_0 + \ldots + n_p = m_0 + \ldots + m_q$

$f_2: n'_0 + \ldots + n'_{p'} = m'_0 + \ldots + m'_q$

$g: n_0 + \ldots + n_p + n'_0 + \ldots + n'_{p'} = m_0 + \ldots + m_q + m'_0 + \ldots + m'_{q'}$

1. Find a derivation for the formula $1 + 1 + 1 + 1 = 2 + 2$. Show that the formulae $2 + 3 = 5$, $1 + 1 + 1 + 1 = 1 + 2$ are not theorems.
2. Find an inference rule r_2 such that if this is added to S the formula $n + m = p$ can be deduced, for all integers n, m, $p > 0$ such that $n + m = p$ is true in the sense of ordinary arithmetic.

   Can the formula $n_1 + \ldots + n_p = m_1 + \ldots + m_q$ be deduced, for any set of integers for which it is true arithmetically?

   Can other formulae be deduced? ? ... (if you answer is yes, give an appropriate second rule r_2) (your system is not sound).
3. What is the effect of removing one of the axioms?
4. What is the effect of adding to the axioms:
   (a) a theorem?
   (b) the formula $1 + 1 = 1$?