WORKING NOTE AC55

Alan Creak
7 July 1986

MECHANICAL DETERMINATION OF FREE-RADICAL REACTION MECHANISMS

(This note preserves in a more accessible form a document which I first wrote while at Derby.)

(I reconstructed it from a rather nasty photocopied typescript in 1999. I corrected one or two obvious spelling or similar mistakes, reformatted the text, and added one short explanatory comment, but it is otherwise unchanged.)

A chemist faced with the problem of determining a mechanism for a free radical reaction starts from a knowledge of the substances originally present in the system (some of which he may know to be consumed during the reaction), a set of reaction products (which may be incomplete), and his general knowledge of reactions and reaction types which have been found in other systems. His object is to select a set of reactions which will account for the observed behaviour and he would prefer to find a small set of known reactions, rather than be forced to invent new reaction types – an example of Occam's razor in action.

His approach to the problem is likely to involve concepts such as "obviously, such-and-such a reaction must be included", and to be in general less than systematic. This is of no importance to the chemist, provided that he gets results, but must be tackled in any attempt to mechanise the process. It is this problem which forms the subject matter of the work described below.

A further item of information which may be available to the chemist has been omitted from the discussion; this is the quantitative behaviour of the system, i.e. the amounts of reactants consumed and products formed. Work is in progress on this topic, which seems to be largely independent of the qualitative considerations discussed here.

THE LOGICAL STRUCTURE OF THE PROBLEM

There are certain necessary conditions which must be satisfied by any reaction mechanism; these will be set out in this section, and briefly discussed if appropriate. They are conveniently written in terms of the calculus of propositions; for those unfamiliar with this branch of mathematical logic, we note that the symbols , &, , and ¬ can conveniently pronounced "implies", "and", "or", and not" respectively.

It is convenient to introduce conventions for notation. Upper and lower case italic letters (e.g. *R*, *c*) will be used to represent physical reactions and compounds. Corresponding letters in heavy type (e.g. ***R***, ***c***) will denote related statements of the form "the reaction *R* occurs (compound *c* exists) in the experimental system". Subscripts will be used as appropriate.

We associate with each reaction R_i the set of its reactants $\{ r_{ij} \}$ and the set of its products $\{ p_{ij} \}$; similarly, we associate with each compound c_i the set of reactions in which it is produced $\{ R_k \mid c_i \ \{ p_{kj} \} \}$ and the set of reactions in which it is consumed $\{ R_k \mid c_i \ \{ r_{kj} \} \}$. We also define three special sets of compounds : the set of compounds known to react, $\{ \rho_i \}$, the set of known products, $\{ \pi_i \}$, and the set of other compounds known to be present initially $\{ \kappa_i \}$.

We may now write down the logical conditions which must be satisfied by the mechanism

1. If a compound is known to react :
   The compound is present in the system, and at least one reaction which consumes it must occur.

$$c_j \ \{ \rho_i \} \quad \boldsymbol{c}_j \ \& \ {}_k \{ \boldsymbol{R}_k \mid c_j \ \{ r_{ki} \} \}$$

2. If a compound is a known product :
The compound is present in the system, and at least one reaction in which it is produced must occur.

$$c_j \{ \pi_i \} \quad \boldsymbol{c}_j \,\&\, {}_k\{ \boldsymbol{R}_k \mid c_j \{ p_{ki} \} \}$$

3. If a compound is known to be present initially :

$$c_j \{ \kappa_i \} \quad \boldsymbol{c}_j$$

4. If a compound is thought to be present but not present initially :
At least one reaction in which it is produced must occur.

$$\boldsymbol{c}_j \quad {}_k\{ \boldsymbol{R}_k \mid c_j \{ p_{ki} \} \}$$

5. If a reaction is thought to occur :
All of the compounds involved in the reaction must be present.

$$\boldsymbol{R}_i \quad \&_j \{ \boldsymbol{c}_j \mid c_j \{ r_{ki} \} \{ r_{ki} \} \}$$

No more information is available from a purely logical analysis of the mechanism. In particular, it is not valid to draw conclusions from the absence of a compound from the known products : the compound may be a reactive intermediate, such as a free radical, which is not sufficiently stable to be detected; or it may be formed in very small quantities, as, for example, the product of an initiation or termination step in a chain-reaction system; or it may be consumed by ill-defined wall polymerisations.

It is thus possible to make one assertion for each compound and each reaction involved in the system, and it is in principle possible, taking into account all known reactions and the given special sets of compounds, to arrive at a composite logical description of the mechanism by postulating all the assertions conjointly. This composite statement can then be rearranged by the methods of the calculus of propositions into its disjunctive abnormal form, each conjunctive component of which must describe a possible mechanism.

PRACTICAL CONSIDERATIONS

From a practical point of view, this programme leaves much to be desired. It is not realistic to include all known reactions in the analysis, so that some preliminary stage in which suitable reactions are selected is essential. We suppose here that this has already been accomplished in some fashion; in practice, the computer programmes developed so far require a manual selection of reactions to start with, and methods for automatic selection of the starting set are under investigation. A second problem is that, in converting the composite statement into its disjunctive abnormal form, expressions with a very large number of terms are generated, some of which must be rejected if the operation is to be performed inside a computer. This problem can be restated as a need to know which of several lines of argument can be neglected when the argument is only partly completed, and has no known satisfactory answer.

The appearance of the problem can, however, be postponed by taking such steps as are available to restrict the number of terms in the growing expression; in order to establish a context for the discussion on this topic, it is convenient at this point to introduce some remarks on the operation of the computer programme.

The element of the logical argument is a set of pairs of computer words called a G-vector, and symbolises a conjunction of assertions about the mechanism – for example, "*a* and *d* are involved in the system, *c* is not involved in the system, *D* is involved in the system, *C* and *E* are not involved in the system, and no information is included on any other reaction or compound", or $\boldsymbol{a}\ \&\ \neg\boldsymbol{c}\ \&\ \boldsymbol{d}\ \&\ \neg\boldsymbol{C}\ \&\ \boldsymbol{D}$

& ¬***E***. Supposing that the reaction set provided included 5 compounds, *a* to *e*, and 7 reactions, *A* to *G*, the G-vector corresponding to the statement above would be :

1	0	0	1	0	-	-	-	-	0	0	0	1	0	0	0	-	-	-	-
0	0	1	0	0	-	-	-	-	0	0	1	0	1	0	0	-	-	-	-
a	***b***	***c***	***d***	***e***					***A***	***B***	***C***	***D***	***E***	***F***	***G***				

Two word-pairs are used (no word contains bits representing both compounds end reactions); a larger G-vector would be used if the l6-bit words of the IBM1130 machine were inadequate. One word of each pair (written at the top) records the asserted statements, and the other the negated statements. Unused bits in a word are always set to zero.

Any statement about the mechanism can be expressed (in disjunctive normal form) as a G-vector, or a disjunction of several G-vectors. The basic statements derived in (1) to (5) above, when so expressed, are called F-vectors, and are simple to construct by inspection of the mechanism. The F-vectors are generated one by one. Each reaction is converted into the corresponding F-vector derived from expression (5), using the identity A B = ¬A B; the F-vector thus contains 2 G-vectors, and is of the form ($\neg \boldsymbol{R}_i$) ($\boldsymbol{c}_1$ & $\boldsymbol{c}_2$ & - - -). When processing a compound, the special sets (known to react, known product, known to be present, and thought to be present) are checked, and an F-vector of type (1) to (4) generated accordingly. These have the forms :

(1) ($\boldsymbol{c}_j$ & $\boldsymbol{R}_1$) ($\boldsymbol{c}_j$ & $\boldsymbol{R}_2$) ....

(2) Similar to (1)

(3) ($\boldsymbol{c}_j$)

(4) ($\neg \boldsymbol{c}_j$) ($\boldsymbol{R}_1$) ($\boldsymbol{R}_2$) ....
(using the transformation previously applied to the reaction F-vectors)

As the F-vectors are generated, they are combined with the growing composite statement; this is again represented by a set of G-vectors, and is called the S-vector. All possible combinations are made each time; thus, if the S-vector contains n G-vectors, and the new F-vector contains m G-vectors :

$$S = S_1 \quad S_2 \quad .... \; S_n$$

$$F = F_1 \quad F_2 \quad .... \; F_m$$

Then the new S-vector, S', could contain up to mn G-vectors :

$$S' = S \;\&\; F = (S_1 \;\&\; F_1) \quad .... \; (S_1 \;\&\; F_m) \quad .... \; (S_n \;\&\; F_m)$$

mn is in fact an upper limit; some of the new G-vectors may represent propositions in which one of the elementary statement is both asserted and denied. Such a G-vector represents a false statement, and is not included in the new S-vector.

The exploitation of this elimination of G-vectors is the only way in which the size of the S-vector can be limited while retaining full logical rigour; it is therefore important so to arrange matters that as many contradictions as possible arise in the earlier stages of the computation. There are two ways in which this can be accomplished. First, it is important to include all the unambiguous experimental data as soon as possible. For each compound known to be present, a term ***c*** is incorporated into every G-vector in the S-vector, and this ensures that all potential G-vectors contain ¬***c*** will vanish. Second, it is important to ensure that reactions and compounds are processed in a suitable order; each reaction contributes G-vectors containing statements such as ¬***R*** or ***c***, while each compound (apart from those known to be

present) contributes terms of the form $\neg \boldsymbol{c}$ or $\boldsymbol{R}$. Elimination can therefore be encouraged by mixing the processing of reactions and compounds, and the scheme adopted throughout is to process a reaction, then those of its compounds which have not already been included. The experimental data are included as soon as possible by arranging the reactions in order so that the compounds known to be present are all included in the first few reactions; the subsequent ordering is guided by the principle that the next reaction for processing should be one involving as many as possible of the compounds which have already been included, again with the aim of encouraging the formation and elimination of contradictory G-vectors.

The importance of these steps is emphasised by considering an example. Suppose that all the reactions were processed first, to be followed by the compounds. Each reaction contributes two G-vectors to the S-vector, and no elimination is possible. Thus, after n reactions have been included, the S-vector will have 2^n terms. This is to be contrasted with the situation actually observed using the scheme described above for a medium-sized reaction mechanism (involving 14 reactions and 16 compounds) : the numbers of G-vectors in the S-vector after including successive reactions were 2, 5, 9, 14, 11, 18, 11, 22, after which the size of the S-vector grew rapidly because only the reactants and two products were specified.