ON THE RECOGNITION OF STRUCTURAL FEATURES: SOME GENERAL PROBLEMS ILLUSTRATED BY HAMILTONIAN PATHS IN POLYHEXES

E.C. KIRBY
Resource Use Institute, 14 Lower Oakfield, Pitlochry, Perthshire PH16 5DS, Scotland, UK

Abstract

Human and machine recognition skills are discussed, though not comprehensively reviewed, and some of the difficulties are illustrated by algorithms written to search for Hamiltonian paths in polyhexes. The most successful strategy for this is based upon the "branching graph", a recently introduced graph-theoretical device which can aid the recognition of edges that are *not* part of a Hamiltonian path. Another, more widely applicable approach that is interesting, although in this preliminary form only a little better than random methods, uses the metaphor of biological evolution, and tries to "breed" and "grow" paths subjected to "natural selection".

1. General introduction

This work has its origin in the practical need for an easy means of transmitting the connectivity of chemical structures from a computer keyboard. For this short-term purpose, a code must be unambiguous but need not be unique, and a convenient technique is to define the structure as a set of connectional differences between it and a chain of the same size [1]. To be sure of achieving the simplest code value, it is of course necessary to identify a Hamiltonian path (one that visits every vertex just once) if one exists (there are sometimes other reasons too for wishing to identify a Hamiltonian path [2–5]). Questions that then arise are: (i) which graphs are Hamiltonian, and how many paths or circuits do they have (the famous and intractable "Hamiltonian problem" [6]), and (ii) how does one find such paths? The distinction is not always sufficiently clearly made. "Enumeration", a term frequently used in mathematical chemistry, has two related but distinct senses in the English language: "the action of ascertaining the number of something" and "the action of specifying seriatim" [7]. Completion of the latter gives the former, but any prior knowledge of the former does not necessarily help the latter. This more practically-oriented task belongs to a class of what can be very subtle recognition problems. Here, it was hoped to refine, improve and cross-check previously reported computer algorithms [2] and to clarify, just a little, some of the broader (and perhaps sometimes rather subjective) issues involved in attempting to mechanise the skills involved.

Since recognition is a crucially important skill possessed by all living organisms, and one that is essential for survival [8,9], it is not surprising that the literature on

© J.C. Baltzer AG, Scientific Publishing Company

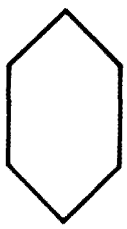
the subject is an immense one, spanning many disciplines (e.g. robotics, computer science, cognitive psychology, neurophysiology, and so on), and no attempt is made here to summarise and document it. In mathematical chemistry, the concept has not been explicitly considered very often (but see Hall and Dias [10] for a recent example), although a few attempts at using graph invariants for recognition purposes have been made, e.g. [11,12]. There is also some semantic overlap with other terms. Examples from the titles of recent papers (not an exhaustive list) are "similarity" [13–15], "comparability" [16], "shape" or "seeing" [17], "perception" [18], and "coding" [19].

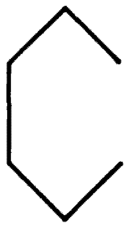
Objects in 3-D space are generally apprehended visually, and visual recognition is perhaps (although not certainly) the most highly developed form in humans. Sensing has several components – such as the exploration of boundaries and contours, colour, reflectivity and degree of light scattering – from which we infer the size, shape and texture of objects in everyday life. The boundaries of objects and their associated visual discontinuities seem to play a particularly important role [20,21]. We do not yet understand *how* we recognise objects. Thus, Longuet-Higgins [22] recently noted that: "It seems fair to say that we simply do not know how the shapes of three-dimensional objects are represented in the long-term memory, how these representations are established in the first place or how they are deployed in the task of visual identification . . .".

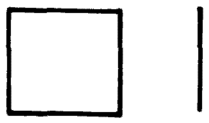
There is a temptation to liken the brain to a computer (and clearly there are fruitful similarities), with the implication that with the right software design, *any* skill can be "taught" to a computer. However, the philospher Searle [23] has sceptically remarked upon a persistent tendency to interpret brain function in the vocabulary of current technology, and the ideas that the brain is some kind of serially processing digital computer, or an extended network, are merely recent examples from a succession of metaphors that have been used. It is certainly true that many human skills of generalised intelligence have not proved easy to mechanise [21]. In particular, it is the combining of "common sense" with the imaginative kinds of skill of which recognition seems typical that seems to be very difficult. Nevertheless, despite such reservations, it is always worth asking whether even quite simple computer methods can be useful for a given task.

2. Computers and the recognition of graphs

Subjectively at least, and for a limited range of problems which perhaps are endowed by our evolutionary history [8,20], there are some striking differences in our behaviour during recognition and how a computer has at present to be programmed to do it. Consider (1), (2) and (3) for example. The difference between them to the eye is immediate, and apparently no calculations or logical comparisons are needed, and the same holds true for much more complex objects such as facial images. It is of course important to remember that this feeling may be an illusion; the steps


(1)


(2)


(3)

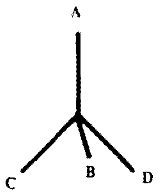
may merely be invisible to us, just as low-level machine instructions are to a high-level language programmer, and not *experienced* in this way. It is difficult to construct a really satisfactory model in computer memory. The traditional adjacency matrix (or its condensed form as a list of edges or a connection table) of graph theory seems intuitively clumsy to use. In part, this is so because (ignoring any symmetry-induced redundancies) there are $N!$ possible equivalent adjacency matrices for any graph on N vertices, so that in order to specify a unique one, special rules must be written, e.g. [24,25]. Various other coding methods (e.g. [13,26–28]) work well, but some have restricted application.

In the case of (1)–(3), the difficulty of a multiplicity of adjacency matrices does not arise because there are no branches. However these graphs are labelled, they can be identified by tracing through, checking for connectedness, while counting numbers of vertices in each valency class. So it is useful to bear in mind that paths are much easier to recognise than branched structures, although there still seems to be no way with a conventional computer program to confer the ability to "see" the graph as a whole. A program must invariably behave like an animal confined to walking the edges of the graph, unable to look in at it as a whole from "outside".

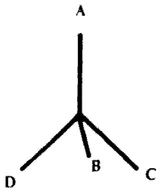
It should also be noted that for the human observer the problem of distinguishing (1)–(3) has been helped by the use of a conventional geometric framework. If, for example, we are faced with an already drawn graph that is not a straight line drawing, but is represented by a tangled bundle of string, then we are apt to be slow, and much less effective than a "blind" computer program in recognising it. Another task at which we are often slow is the trivial one of establishing the size of a ring that has more than about eight vertices. However clearly it is drawn, we usually have to count the vertices rather carefully to avoid mistakes, whereas the machine will have this information instantly available in its adjacency matrix model.

The difference between (4) and (5) when arranged in a tetrahedral configuration can, although this often needs a measurable time for mental effort, be recognised easily by sight, but is invisible in an adjacency matrix unless some arbitrary convention for labelling branching vertices with their chirality is adopted.

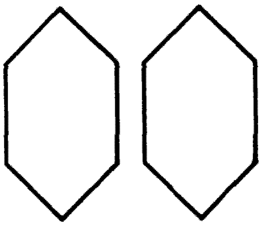
Consider also (6) and (7); we can again see the difference immediately, but an adjacency matrix does not have this information, and additional and separate



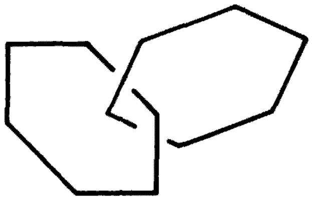
(4)



(5)



(6)

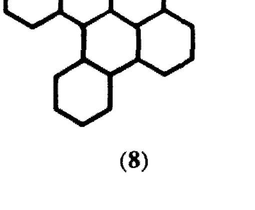


(7)

concepts like link numbers and knot theory [29] must be invoked. (Note: hydrocarbon structures like (7) would be chemically feasible with larger rings only, but that is irrelevant here.)

3. Hamiltonian paths and circuits in polyhexes

In the light of the foregoing discussion, it is of interest to consider Hamiltonian paths [30, 6] in polyhexes. The problem of identifying these paths retains a fascination, because on systems of only moderate size it can be surprisingly difficult to see one. There are a number of Hamiltonian paths in the polyhex (**8**) for example, and you the reader may see one immediately. Nevertheless, in recent work two scientists originally formulated structure (**8**) as a possible example of a completely non-Hamiltonian polyhex! The skill varies among individuals, and tends to improve somewhat with practice.



(**8**)

Here then is a recognition task that is not difficult merely because of the tedium of processing a long list, but is also inherently difficult because of the large number of confusingly similar but invalid paths within a given structure. Can a computer algorithm provide assistance?

3.1. A SEARCH FOR HAMILTONIAN PATHS USING BRANCHING GRAPHS

It is important to appreciate that the act of recognition involves two things: the object, and *what of significance the object excludes*. An analogy may be drawn to the activity of sculptors who, at least when working from a block, must know what material needs to be removed [9]. This is the strategic basis of the use of "branching graphs" (see appendix for definitions).

There is a one-to-one correspondence between the 2-factors of a polyhex (P) and the 1-factors of its branching graph $B(P)$ [31], and erasure of the edges of a branching graph 1-factor from the polyhex leaves a 2-factor. Figure 1 shows an example. This enables one to enumerate the 2-factors of P by enumerating the 1-factors of $B(P)$, and so to find any Hamiltonian circuits (which are connected 2-factors) that exist. For some other work on 2-factors, see [32–37].

The concept of 1-factors as being a subclass of more generalized "k-branch-factors" enables any subgraphs with just two disjoint branching vertices to be identified by a similar process of edge erasure. A k-branch factor is a set of disjoint edges that accounts for all but k vertices (see appendix). Derived from a polyhex, the analogous "factor" – a doubly branched subgraph – can have one of only two possible forms, and both are traceable in an obvious manner provided that the "factor" is connected, so that Hamiltonian paths as well as circuits can be identified (exemplified by perylene and pyrene in fig. 1). If the only possible k-branch factors have $k > 2$, then the polyhex is untraceable. (Note: it is possible to formulate rules for constructing many Hamiltonian and non-Hamiltonian polyhexes [2] but they are rather clumsy to use in practice.)

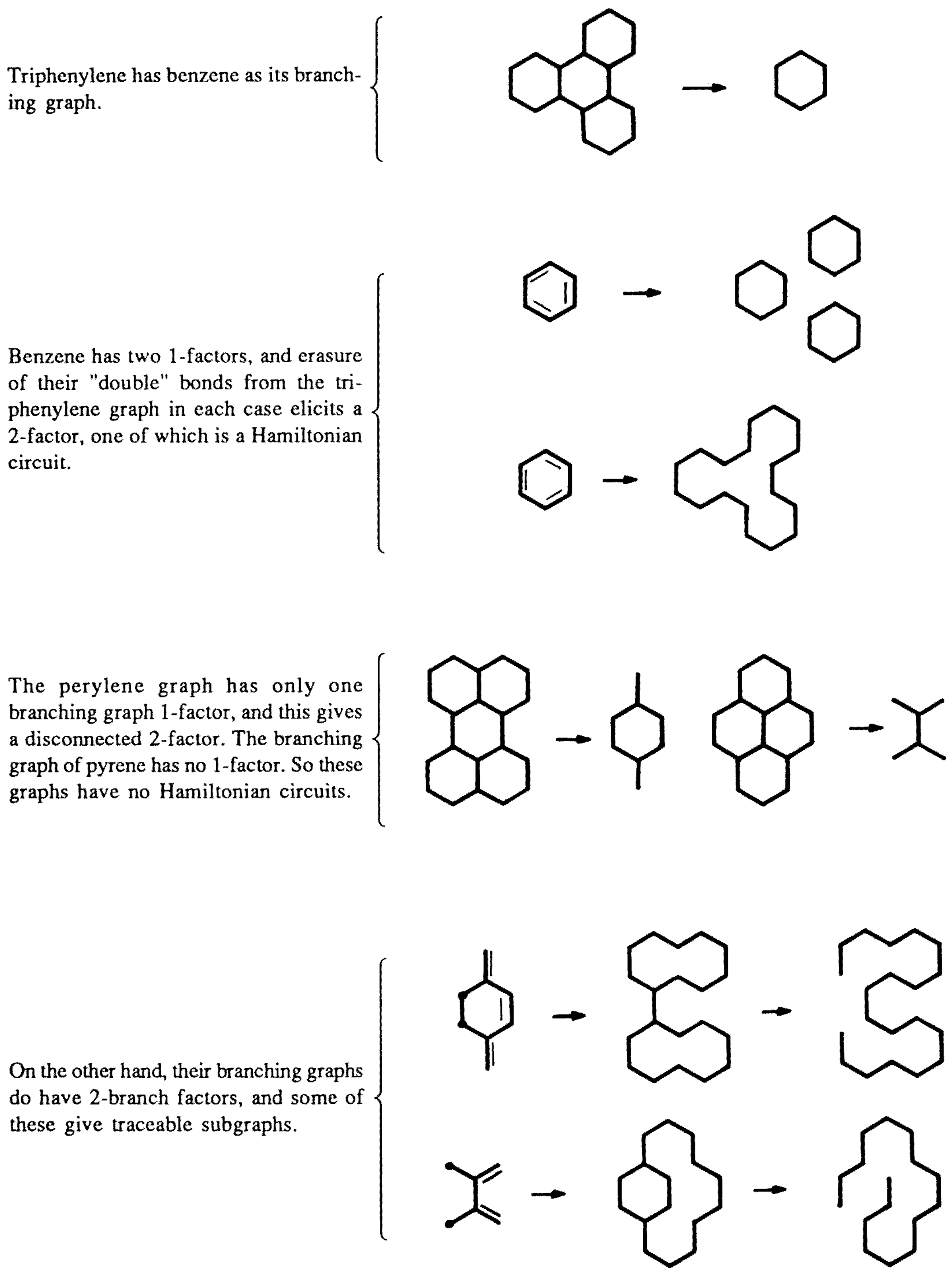
When this technique is applied as a computer algorithm, a connection table for the branching graph is derived from that of the parent polyhex and searched for k-branch factors [38]. Hamiltonian paths or circuits are stored as arrays of vertex numbers.

3.2. OTHER METHODS OF SEEKING HAMILTONIAN PATHS

Because Hamiltonian paths can be so difficult to see, it is highly desirable that some independent alternative algorithms be developed to provide a check on results.

3.2.1. *Random searching*

Apart perhaps from the statistically hopeless method of guessing at completely random sequences of vertices, this is the simplest approach. A starting vertex is



Triphenylene has benzene as its branching graph.

Benzene has two 1-factors, and erasure of their "double" bonds from the triphenylene graph in each case elicits a 2-factor, one of which is a Hamiltonian circuit.

The perylene graph has only one branching graph 1-factor, and this gives a disconnected 2-factor. The branching graph of pyrene has no 1-factor. So these graphs have no Hamiltonian circuits.

On the other hand, their branching graphs do have 2-branch factors, and some of these give traceable subgraphs.

Fig. 1. The recognition of a Hamiltonian path or circuit following recognition of edges (identified via the branching graph) that are irrelevant (see appendix for definitions).

selected at random and a path traced from it through the connection table. Whenever a branching vertex is encountered, a choice of direction is made at random. The path is continued until no further progress is possible without revisiting an already used vertex. If at this point all vertices have been accounted for, then a Hamiltonian path has been found, and the possibility that this may be part of a Hamiltonian circuit can quickly be checked by seeing whether the last vertex is adjacent to the first one. Otherwise, the trial is abandoned and another one started.

As one might expect, this method is quite effective for small systems, but the time needed increases very rapidly with increasing size.

A modification in which failure is followed by reversal to the last branch, seeking for a new route, does not improve matters, presumably because the program often wastes too much time checking clusters of paths that all emanate from an early unsuitable branching point.

3.2.1. An "evolutionary" search for Hamiltonian paths

In this, subgraphs of a polyhex are treated as "animals" that can grow and reproduce, by borrowing as a metaphor the workings of biological evolution [39]. Several approaches to implementation were tried. In the most successful so far, the set of edges is regarded as a set of "cellular automata" (see, for example, Wilson [40]) which may be switched on or off. A "parent", consisting of a random subgraph, "breeds" a certain number of copies of itself, each of which is vulnerable to some random switching on or off of edges ("genetic mutation"). These "children" then all grow as far as possible, becoming "adults", according to set rules, and selection criteria are applied to find a new parent for breeding. As before, the essential structure is held as a connection table, and edges are "switched off" by giving the appropriate vertex numbers in the table a negative value. The scheme is illustrated with a single imaginary example in fig. 2.

Even these preliminary and little-optimised results (table 1) do seem rather better than those from random effects alone for systems beyond a certain size, although further optimisation is desirable. For small structures, a random search is more effective, because less complicated programming overheads are needed. The most substantial difficulty is probably the devising of suitable selection criteria, and this raises an interesting question: how can a subgraph best be characterised, if indeed it can at all, as being more or less "nearly Hamiltonian" with respect to the graph as a whole? At first sight, it might appear that the one with the longest unbranched path should be selected. This is not necessarily so however, for a longer path may have assumed a geometric configuration that inhibits further growth, and be less suitable than some more branched candidate. The best one must be a subgraph that requires fewest changes to convert it to a Hamiltonian path, but this is difficult to quantify in any simple manner until it can be done (redundantly) in retrospect. In this trial, a maximum value of the expression $V_2/(V_3 + 1)$ was sought ($+1$ in the denominator is needed to avoid the possibility of dividing by zero).

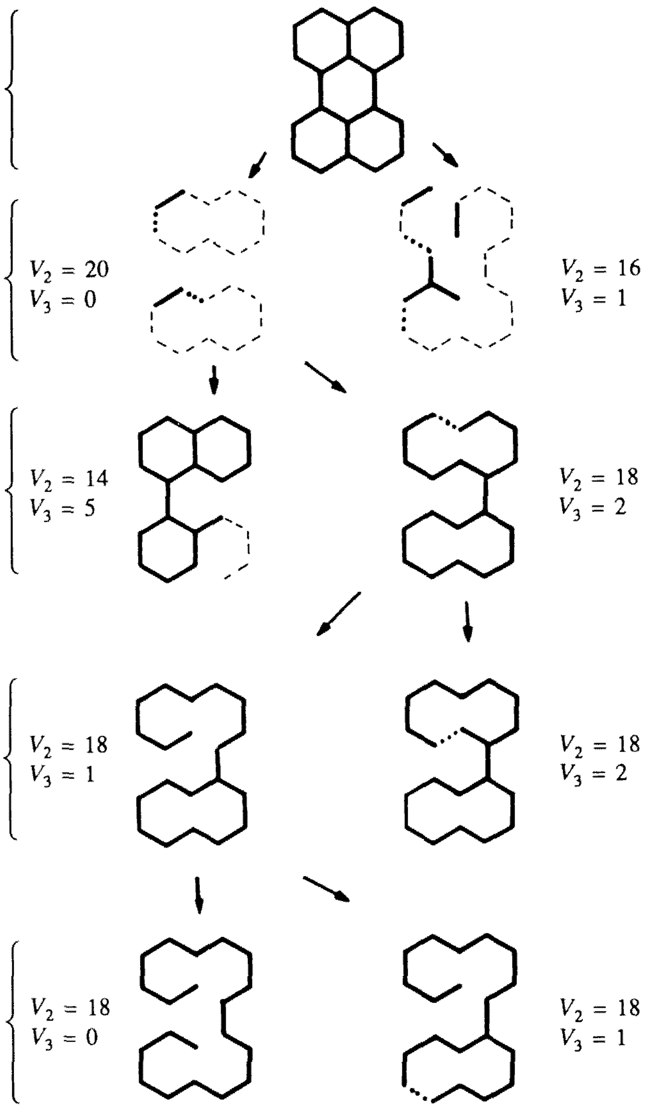
Polyhex P for checking.

Two children: suppress all P's edges except for a few chosen at random. Allow these to grow as far as possible in accordance with the rules. Choose the resultant graph with the most degree two and the fewest degree three vertices to continue with as the next parent.

Two grandchildren: obtained by copying with mutations, followed by growth. Repeat the selection.

Repeat the sequence

And again: here, one of the "mutations" is a Hamiltonian path; the routine terminates.



————	Edges of a "parent" graph after random changes have been made.
— — — —	Edges generated by successively adding terminal edges to terminal vertices of the above.
· · · ·	Edges generated by bridging pairs of terminal vertices after the previous process if complete.

Fig. 2. An example of a possible sequence during an "evolutionary" approach to Hamiltonian paths. Note: The selection and growth techniques shown are examples tried from a number of possibilities, but probably they are not optimal.

Table 1

Some rough comparisons of the time to discovery[a)] of a Hamiltonian path or circuit by three search methods (see text for details)

	Random search	Branching graph method	Evolutionary search
A linear polyacene with 10 hexagons:			
Mean	18.45	b)	40.4
Standard deviation	23.43		22.6
Range	2–93		11–96
Sample size	22	22	22
Polyhex (**8**)			
Mean	c)	12.64	2745
Standard deviation		10.73	–
Range		2–41	972–6275
Sample size		22	4

a)Time in seconds, running on a Compaq SLT286 portable computer.
b)Always <0.5 seconds.
c)No success in 5 trials lasting up to 4000 seconds.

It is an attractive feature of the strategy, however, that there does not need to be a perfect ordering. It is sufficient that a high value of the selection parameter for a subgraph be associated with there being required, just more probably, less reconstruction to become a Hamiltonian path. For the same reason, that the method does not depend on any one particular sequence of mathematical operations, it should in principle be widely applicable.

3.3. POLYHEX INPUT AND DISPLAY

A polyhex (in an arbitrary orientation, but with some edges vertical) is encoded by entering the rectangular coordinates of the hexagons. The program assigns a unique number in an arbitrary though systematic way to each vertex of each hexagon as the latter is given. The numbers are stored as the elements of a rectangular matrix in accordance with the (slightly distorted) two-dimensional geometry of the polyhex (cf. [26]). At the same time, a connection table is also built up and stored. A simple but useful representation of the polyhex, or any subgraph of it, can be composed of ASCII characters and screened or printed out for recognition from the information in these two matrices.

4. Program code

The algorithms and routines described here were implemented in compiled Borland TurboBasic v1.0.

Appendix

GRAPH THEORETICAL DEFINITIONS USED

Polyhex: A polyhex is a network of regular hexagons such that any two of them are either disjoint of have a common edge. For some recent introductory reviews of the subject, see [41–44].

Benzenoid: Although the nomenclature is not universal, the term benzenoid is often used to refer to the subclass of polyhexes that have 1-factors.

Hamiltonian path: If a graph has one, this is a path that visits every vertex just once. If such a path can be continued to the starting vertex, then it is a *Hamiltonian circuit*. These graphs can be described as *traceable*, and a graph with a Hamiltonian path (but not a circuit) is sometimes referred to as being *semi-Hamiltonian*. A graph that is *non-Hamiltonian* means here one that has no Hamiltonian path and no Hamiltonian circuit.

1-factor: If a graph has one, it is a set of disconnected edges that can be drawn to include every vertex, so that all vertices are of degree one. It is equivalent to the set of "double" bonds within a Kekulé structure.

k-branch-factor: This is a set of disconnected edges that can be drawn to account for all but *k* of the vertices (where *k* is even). The device is defined by generalising the concept of a 1-factor using, as an analogy, the relationship in organic chemistry between principal resonance structures in general, and Kekulé structures in particular. So, using this nomenclature, a 1-factor is the same as a 0-branch-factor.

2-factor: If a graph has one, this is a set of one or more rings that can be drawn to include every vertex, so that every vertex is of degree two. If a 2-factor consists of only one ring, it is called a Hamiltonian circuit.

Branching graph: The branching graph of a polycyclic graph is a special subgraph. Each vertex of the parent graph G appears in its branching graph $B(G)$ if and only if it is a branching vertex. Each edge of G appears in $B(G)$ if and only if it connects two such branching vertices [2,45].

References

[1] E.C. Kirby, J. Math. Chem. 1(1987)175.

[2] E.C. Kirby, J. Math. Chem. 4(1990)31–46.

[3] R.B. Mallion, private communication (12th April 1989).

[4] J. Gayoso and A. Boucekkine, Comp. Rend. Acad. Sci. (Paris) C272(1971)184.

[5] R. McWeeny, Mol. Phys. 1(1959)311.

[6] N.L. Biggs, E.K. Lloyd and R.J. Wilson, *Graph Theory 1736–1936* (Clarendon Press, Oxford, 1976).

[7] *The Shorter Oxford Dictionary*, 3rd ed. (Guild Publ., London, 1988).

[8] Charles R. Darwin, *On the Origin of Species by Means of Natural Selection* (1859).

[9] J. Bronowski, *The Ascent of Man* (British Broadcasting Corporation Publ., London, 1973).

[10] G.G. Hall and J.R. Dias, J. Math. Chem. 3(1989)233.
[11] M. Randić, Theor. Chim. Acta 62(1983)485.
[12] T. Okuyama, Y. Miyashita, S. Kanaya, H. Katsumi, S. Sasaki and M. Randić, J. Comput. Chem. 9(1988)636.
[13] W.C. Herndon and S.H. Bertz, J. Comput. Chem. 8(1987)367.
[14] P.G. Mezey, J. Math. Chem. 2(1988)299.
[15] M.A. Johnson, J. Math. Chem. 3(1989)117.
[16] D. Bonchev, V. Kamenska and O. Mekenyan, Int. J. Quant. Chem. 37(1990)135.
[17] F. Harary and P.G. Mezey, J. Math. Chem. 2(1988)377.
[18] G.M. Downs, V.J. Gillet, J.D. Holliday and M.F. Lynch, J. Chem. Inf. Comput. Sci. 29(1989)172.
[19] R.C. Read, J. Chem. Inf. Comput. Sci. 23(1983)135; 25(1985)116.
[20] J. Graham Kerr, Trans. N.E. Coast Inst. Eng. and Shipbuilders 35(1919)256; Nature 149(1942)221; Naut. Mag. (1942) June 22.
[21] M.A. Boden, *Artificial Intelligence and Natural Man* (Harvester Press, 1984).
[22] C. Longuet-Higgins, Nature 343(1990)214.
[23] J.D. Searle, BBC Reith Lectures (1984); *Minds, Brains and Science* (Penguin Press, London, 1989).
[24] M. Randić, J. Chem. Phys. 60(1974)3920.
[25] J.B. Hendrickson and A.G. Toczko, J. Chem. Inf. Comput. Sci. 23(1983)171.
[26] J.V. Knop, W.R. Müller, K. Szymanski and N. Trinajstić, *Computer Generation of Certain Classes of Molecules* (SKTH/Kemija u industriji, Zagreb, 1985).
[27] J. Cioslowski, J. Comput. Chem. 8(1987)906.
[28] M. Randić, Croat. Chem. Acta 59(1986)327, and references therein.
[29] D.W. Sumners, J. Math. Chem. 1(1987)1, and references therein.
[30] T.P. Kirkman, Roy. Soc. London 146(1856)413 (cited in ref. [6]).
[31] I. Gutman and E.C. Kirby, MATCH 26(1991), in press.
[32] H. Hosoya and T. Yamaguchi, Tetrahedron Lett. (1975)4659.
[33] N. Ohkami, A. Motoyama, T. Yamaguchi, H. Hosoya and I. Gutman, Tetrahedron 37(1981)1113.
[34] S. El-Basil, P. Krivka and N. Trinajstić, Croat, Chem. Acta 57(1984)339.
[35] T.G. Schmalz, G.E. Hite and D.J. Klein, J. Phys. A, Math. Gen. 17(1984)445.
[36] J.R. Dias, Nouv. J. Chim. 9(1985)125.
[37] J.V. Knop, W.R. Müller, K. Szymanski and N. Trinajstić, J. Comput. Chem. 7(1986)547.
[38] E.C. Kirby, Comput. Chem. 9(1985)155.
[39] R. Dawkins, *The Blind Watchmaker* (Longman, 1986; Penguin Books, London, 1988).
[40] G. Wilson, *The life and times of cellula automata,* New Scientist (8th October, 1988), p. 44.
[41] I. Gutman and S.J. Cyvin, *Introduction to the Theory of Benzenoid Hydrocarbons* (Springer, Berlin, 1989).
[42] J.V. Knop, W.R. Müller, K. Szymanski, S. Nikolić and N. Trinajstić, in: *Computational Chemical Graph Theory*, ed. D.H. Rouvray (Nova, New York, 1990).
[43] N. Trinajstić, J. Math. Chem. 5(1990)171.
[44] J.R. Dias, *Handbook of Polycyclic Hydrocarbons* (Elsevier, Amsterdam–New York, Part A 1987; Part B 1988).
[45] E.C. Kirby, J. Chem. Soc. Faraday Trans. 86(1990)447.