How Can Parallel Algorithms Help To Find New Sequential Algorithms?

Zoltán M. Nagy

Törökvész ut 143/a, Budapest, Hungary

Received August 27, 1992

A logical computing model which is a special case of the relaxation method is described, and some well-known algorithms are reformulated using this model. With a combination of a ring perception and a labeling method, new algorithms are created for checking the isomorphism of molecules and for substructure searching.

INTRODUCTION

During a search for new structure handling problems, various models were studied. Instead of modeling the human brain or creating a new general parallel computing model, an attempt was made to find the simplest network which is able to compute specific features of a graph (or molecule). It was found that for most of the structure handling problems the simplest network has the same topology as the molecule. These networks are not invoked as actual hardware but as a logical construction. In this sense they proved to be helpful in studies of new algorithms and in efforts to simplify and speed up the old ones.

The well-known Morgan algorithm,[1] which generates a unique name for a molecule, is considered. The first part of this algorithm classifies the atoms of the molecule. At the beginning each atom has the number of its neighbors as a starting label, and in each step the new label of an atom is the sum of the labels of its neighbors. If there were a network having the same topology as the molecule, as shown in Figure 1, with a simple processor at its nodes, each one able to sum the labels of its neighbors, then the new labels could be calculated in one step.

More generally, a model for each molecule can be defined as shown in Figure 2. The initialization function sets the starting value of the node functions; the node function calculates the new value of the node using the previous value of the neighboring nodes, including itself. The computing of this model is controlled by a supervisor with clock signals, and the supervisor is able to check the stop condition for the network. For the Morgan algorithm this process is illustrated in Figure 2.

One of the most important advantages of this model is that it forces one to focus on the node function instead of becoming lost in the details. In most cases it is straightforward to create the equivalent sequential algorithm which is a special case of the relaxation technique.[2] In the following sections examples will be shown of the node functions related to some structure handling problems.

MINIMAL RING SIZE OF AN ATOM

Figure 3 shows the initial value and the node function used to calculate the minimal ring size of atom *a*. The node function is a simple logical OR. If the node is not node *a*, the algorithm then checks the values of its neighbors: the new value will be 1 if at least one of them is 1; otherwise it will be zero. The process will terminate if the value of node *c* or *d* is 1. The size of the minimal ring is the number of steps + 2. The

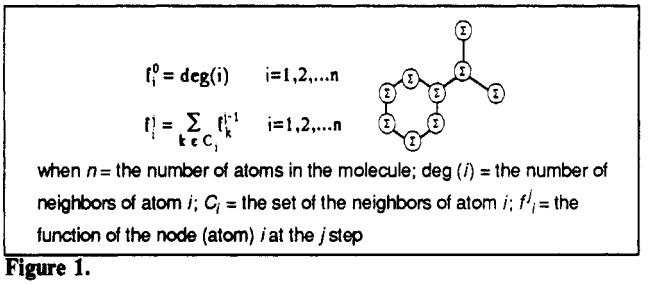


Figure 1.

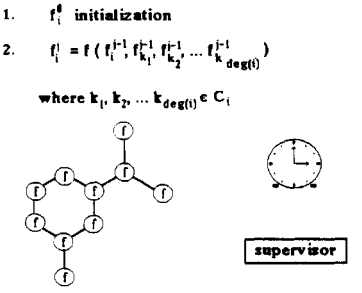


Figure 2.

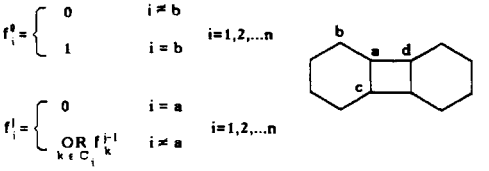


Figure 3.

computing requires one bit storage for each node. The minimal ring size is used in HTSS[3] to classify the atoms.

COORDINATION GENERATION

The node function, as shown in Figure 4, does not solve the complex problem of the coordinate generation. It is just an example which tries to find the minimal energy level for the molecule. It uses 2D coordinates and assumes that all of the normal bonds have a length of 1. The supervisor checks the collisions and, if necessary, makes some modification of the coordinates. Each node is able to store a coordinate and able to create the vectorial sum of the vectors pointing from the node to its neighbors. In the case of the atoms with two neighbors it also calculates the normal vector of the vector lying between its two neighbors. In the case of the atoms with a single connection the node function calculates the length of the vector pointing to its neighbor. The process can terminate if there is no collision or the number of steps exceeds a predefined limit. In order to achieve faster termination one

0095-2338/93/1633-0542$04.00/0

$$f_i^0 = (i, i) \qquad i = 1, 2, \ldots n$$

$$f_i^l = \begin{cases} \frac{1}{|\vec{r}_{ik}|}\vec{r}_{ik} & k \epsilon C_i,\ \deg(i) = 1 \\ \frac{1}{2}\vec{r}_{kl} + \frac{1}{2}n(\vec{r}_{kl}) & k,l \epsilon C_i,\ \deg(i) = 2 \\ f_i^{l-1} + \frac{1}{2}\sum_{k \epsilon C_i} \vec{r}_{ik} & 2 < \deg(i) \end{cases}$$

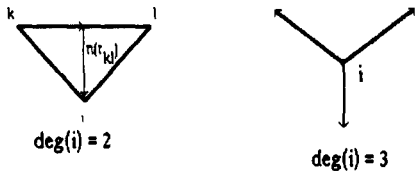


Figure 4.

$$f = (\bar{V}, \bar{W}) = ((v_1, v_2, \ldots v_n), (w_1, w_2, \ldots w_n))$$

$$f_i^0 = (\bar{V}_i^0, \bar{W}_i^0) \quad \text{where} \quad \begin{cases} v_j = \begin{cases} 1 \text{ if } j=i \\ 0 \text{ if } j \neq i \end{cases} \\ w_j = 0 \end{cases} \qquad j=1,2,\ldots n$$

$$f_i^l = (\bar{V}_i^l, \bar{W}_i^l) = (((\underset{k \epsilon C_i}{OR} \bar{V}_k^{l-1}) \text{ AND } (\text{NOT } \bar{W}_i^{l-1})), \bar{V}_i^{l-1})$$

Atom i has a 2j member ring if $\exists$ k,l ϵ C_i for which $\bar{V}_k^{j-1}$ AND $\bar{V}_l^{j-1} \neq \bar{0}$

Atom i has a 2j-1 member ring if $\exists$ k ϵ C_i for which $\bar{V}_k^{j-1}$ AND $\bar{V}_i^{j-1} \neq \bar{0}$

Figure 5.

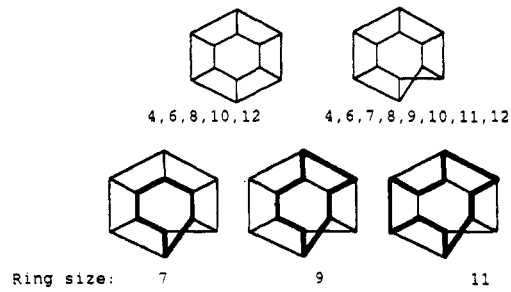


Figure 6.

can set better initial values. Although this function is simple, its use is not recommended in high quality algorithms because it does not generate nice coordinates for complicated structures.

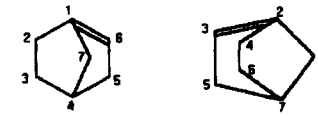
ANALYSIS OF THE RING SYSTEM OF A MOLECULE

To check isomorphism, we need all the possible rings (not only the minimal rings) of the molecule. The node function in Figure 5 is able to calculate all of the possible ring sizes for each atom in the molecule. It is a very useful characteristic when the isomorphism of two molecules is being checked. Using this node function all of the nodes should be able to store $2n$ bits. The function performs simple logical operations on bit vectors. This process terminates in $n/2$ steps, and the supervisor stores the ring sizes for each atom. The power of this feature is shown in Figure 6. These two molecules can not be distinguished using the minimal ring size, but they are very different in the scope of the sizes of all possible ring sizes. Figure 6 shows some ring sizes which are present only in the second molecule.

CHECKING THE ISOMORPHISM OF TWO MOLECULES

The following node function is able to pack the information about the topology of the molecules into each atom

$$f_i^0 = (a, b_1, b_2, \ldots, b_{\deg(i)})$$



step 0.

1.)	(C,s,s)	(2,3,5,7)	(C,s,s)	(1,4,5,6)
2.)	(C,s,d)	(6)	(C,s,d)	(3)
3.)	(C,s,s,s)	(4)	(C,s,s,s)	(7)
4.)	(C,s,s,d)	(1)	(C,s,s,d)	(2)

step 1.

1.)	(1,3)	(3)	(1,3)	(6)
2.)	(1,4)	(2)	(1,4)	(4)
3.)	(2,3)	(5)	(2,3)	(5)
4.)	(3,4)	(7)	(3,4)	(1)
5.)	(1,4)	(6)	(1,4)	(3)
6.)	(1,1,1)	(4)	(1,1,1)	(7)
7.)	(1,1,2)	(1)	(1,1,2)	(2)

step 3.

1.)	(±2,±6)	(3)	(±2,±6)	(6)
2.)	(±1,±7)	(2)	(±1,±7)	(4)
3.)	(±5,±6)	(5)	(±5,±6)	(5)
4.)	(-6,-7)	(7)	(-6,-7)	(1)
5.)	(±3,±7)	(6)	(±3,±7)	(3)
6.)	(±1,±3,-4)	(4)	(±1,±3,-4)	(7)
7.)	(±2,-4,±5)	(1)	(±2,-4,±5)	(2)

\+ indicates an even member ring,
\- indicates an odd member ring

Figure 7.

where a is the atom type of atom i and b_k is the bond type to the kth neighbor.

$$S_j = (f_1^j, f_2^j, \ldots, f_k^j) \qquad f_1^j < f_2^j < \ldots < f_k^j$$

$$f_i^j = (F_1^{j-1}, F_2^{j-1}, \ldots, F_{\deg(i)}^{j-1}) \qquad F_i^{j-1} = \text{index of } f_i^{j-1} \text{ in } S_j$$

Here the nodes are initialized with the atom type and the bond types. The supervisor sorts these arrays and assigns the index of an array in the sorted list to the nodes. It is only an abbreviation and one could use for example a hashing instead of the sorting. The node function uses this index to calculate the next value of the node. The process can be terminated at the $(n/2)$th step, because this is sufficient to check the isomorphism. A node at the jth step has information about its j-radius environment, and one could reconstruct the atom types and the bond types within this environment but could not reconstruct the ring system of the molecule. To get information about the ring system of the molecule, one must combine this node function with the previous one. These two functions together form a powerful function for isomorphism checking. This function is applied in parallel for the two molecules. Beside controlling the process for each molecule, the supervisor compares the values of the nodes after each step. If there is a difference between the set of the node values, then the two molecules are not isomorphic. This calculation is shown in Figure 7. Step 2 is omitted because it is the same as step 3 but without ± signs. The first number is the index after sorting the original values of the node function. The values within the parentheses at step 0 are the atom type and the bond types and, in subsequent steps, the node values of the neighbors at the previous level. The numbers within braces are the labels of the atoms having the same node value. At the end of the algorithm the result will be the mapping between the two molecules, provided they are isomorphic.

SUBSTRUCTURE SEARCH

A similar method can be used for substructure searching, but, in this case, the node function for the substructure has

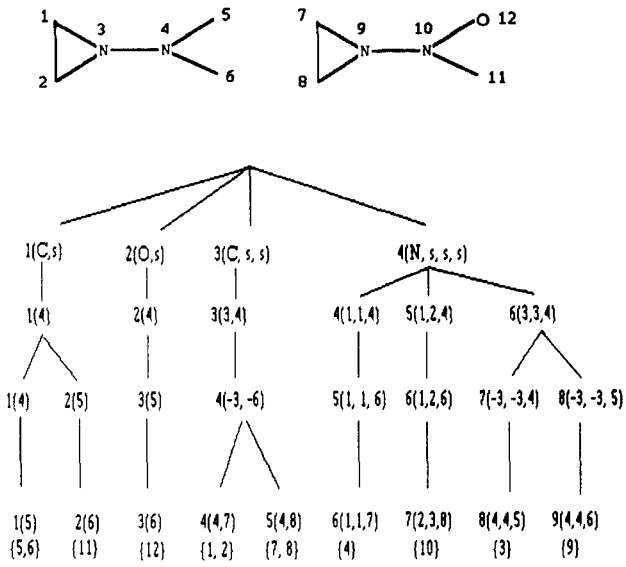


Figure 8.

to include all of the possible values instead of a single value. The corresponding atoms in the molecule must have the same size of ring as in the substructure, but they can have more different rings. If we want to search substructures in a set of molecules (in a database) then we could calculate the node function simultaneously for all of the molecules. The computation of the node function does not depend on the substructures, so we can calculate the node function once for each molecule in the database and we can store them for future use. If we collect all of the atoms throughout the whole database having the same value, then we shall get a tree structure similar to that shown in Figure 8. The *j*th level of this tree represents the *j*th step, and a node of the tree represents a set of atoms having the same node function value. We can build up this hierarchic tree once for the whole database, and we can use it several times for both structure and substructure search. During the substructure search we calculate the node value for the atoms of the substructure and compare it with the nodes of the tree. An atom of the substructure belongs to a node of the hierarchic tree if the value of the tree node exists in the set of the possible values of the atom. At the bottom of the tree—where we store the atoms of the database—we shall get the mapping between the atoms of the substructure and the atoms of the database. The process of search for a simple substructure is shown in Figure 9. Unfortunately the process described above is not always adequate for a correct substructure search because it can produce false hits. If the substructure contains ring(s), then after walking through the tree it will be necessary to check the mapping for those molecules which have larger ring systems

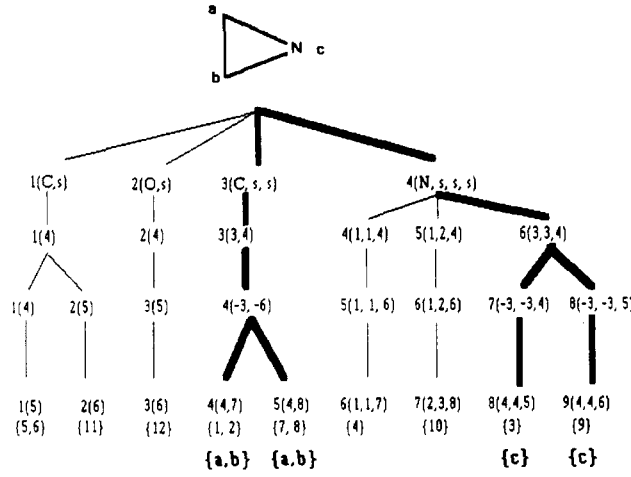


Figure 9.

than can be found in the query structure. This is not a serious penalty because when the bottom of the tree is reached, the mapping between the atoms of the substructure and the atoms of the molecules in the database is available. If during a search it is found that an atom of the query structure cannot be put into any node, then this atom must be replaced by a free site. As a result of this we shall get a set of molecules having common parts with query structure. This is a special similarity search.

OTHER PROBLEMS

An interesting question concerns the kind of features of a molecule that can be calculated if the supervisor only gives the clock signal and checks the termination. Some very simple problems such as calculation of the number of nodes or the number of connections in the network require a quite complicated node function and several steps, but calculation of the molecular diameter (the longest distance), for example, requires a very simple node function. A further study could find out what features can be computed with only a limited memory (which is a predefined constant and does not depend on the size of the molecule) in the nodes.

REFERENCES AND NOTES

(1) Morgan, H. L. The Generation of a Unique Machine Description for Chemical Structures-A Technique Developed at Chemical Abstracts Service. *J. Chem. Doc.* **1965**, *5*, 107–113.
(2) Hummel, R. A.; Zucker, S. W. On the foundations of relaxation labeling processes. *IEEE Trans. Pattern Anal. Mach. Intell.* **1983**, *5*, 267–287.
(3) Nagy, Z. M.; Kozics, S. Veszpremi, T.; Bruck, P. Substructure Search on Very Large Files Using Tree-Structured Databases. In *Chemical Structures*; Warr, W. A., Ed.; Springer-Verlag: Berlin, Heidelberg, 1988; pp 127–130.