PROBLEM 7

MODIFICATIONS TO THE LANGUAGE "ORION"

1. INTRODUCTION

Interactive Engineering Pty. Ltd. of Parramatta, NSW, has developed a mathematical modelling language called ORION (Brander and Royle, 1985). ORION is a member of the class of languages which includes PROLOG (Sammut and Sammut, 1983a,b), in that it has both a declarative and procedural semantics. The declarative semantics consists of a high level description of the set of mathematical and logical relationships in the model, together with a representation of these relationships in the form of a network. The procedural semantics consists mainly of a basic strategy for calculating desired unknown parameters of the model given a set of known parameters by searching the computational network. In addition, there are a number of specialized evaluators which are called in when the basic evaluator is unable to proceed. The specialized evaluators include, among others, an iterator and a simultaneous equation solver.

2. STATEMENT OF THE PROBLEM

The problem posed to the Mathematics-in-Industry Study Group is the specification of a specialized evaluator which can be used when the computation is indeterminate in detail, but does in fact contain enough information to produce a solution. For example, suppose the model consists of the two equations

$$A = B + C$$

$$B + C = 5$$

and the command is given to find A. There is not information to determine either B or C, but it is clear by inspection that A has the value 5. A list of six examples was provided by Interactive Engineering, involving combinations of arithmetic and logical operations.

3. BACKGROUND

A model in ORION is represented as a network. Each variable or constant is represented by a terminal node. Terminal nodes are connected by computational links expressing the mathematical relationships between them. A given variable is represented by only one node. If it participates in a number of equations, it will have several computational links: one for each equation.

When computation of a variable is requested, a breadth first search is started at the node of that variable. When the search reaches a computational link with more than two branches, the search is split and proceeds along both branches. When the search reaches a terminal node with a definite value (constant or variable with a value assigned), the value is sent back along the path. When two or more values arrive at a computational link, the computation is performed and the calculated value sent along. For example, if a link expresses X + Y = Z and X and Z are passed through it, the output at the Y arm is correctly computed as Z - X. This is illustrated in Figure 1.

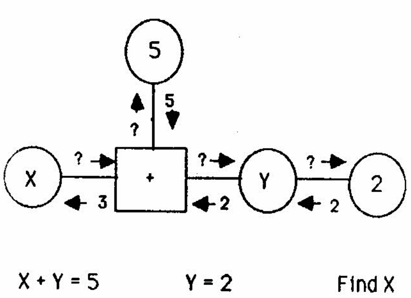


Figure 1: Computation in Orion

The problem presented is expressed in the network representation in Figure 2. The value 5 cannot pass the plus link, because there is only one arm determined. The search is said to be blocked at that point.

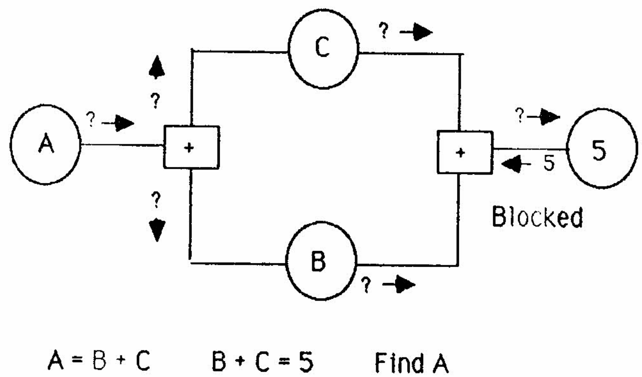


Figure 2: A Blocked Computation

A number of problems other than the one submitted were raised by the group, including:

1. Computational stability. What does the system do about situations such as

$$A = \frac{B}{C - D}$$

where B and C − D are of order 1, but C and D are of order 10^9, or a system of simultaneous equations which is nearly degenerate?

2. What happens in a nonlinear system with more than one solution?

3. How does it recognize and deal with under or over-determined systems?

4. Can we guarantee termination of the computation?

5. What are the convergence characteristics of the iterative techniques in different kinds of problems?

6. Is there a comprehensive trace-back facility so it is possible to follow how the solution was arrived at?

7. Can we characterize (and automatically recognize) classes of problems for which the system as stated provides reliable and comprehensible solutions?

4. SOLUTION

As it turned out, there was sufficient interest and expertize in the problem as stated that some progress could be made. The main body of theory found to be relevant was automatic algebra such as that used in the MACSYMA system, automatic theorem proving, and the theory of graph isomorphisms.

Two general points were made:

1. It is not possible to use floating point values as constants in automatic algebra. Any constants used must be integers.

2. Small logic problems can be solved by exhaustive enumeration of truth tables. This is very simple and could be practical for systems with up to 5 or 6 variables.

A general strategy for solving the problem was suggested.

1. Detect the situation.

2. Identify a portion of the network as a candidate for simplification.

3. Perform simplifications.

4. Recognize the solution.

The first point was not addressed.

Points two and four could be addressed using the theory of algebraic dimension. An algebraic system has a dimension which is the number of degrees of freedom in selection of variables. A zero dimensional system has a finite number of solutions. A one-dimensional system has a one-parameter family of solutions, and so on. A path in the network through which a solution can be sent must represent a zero-dimensional system.

If a computation is blocked and there is any hope of solution, there must be more than one potential path through. It would be possible to classify subsets of these paths by dimension, then choose subsets for simplification in sequence of increasing dimension. A solution could be recognized when the dimension of a subsystem has been reduced to zero.

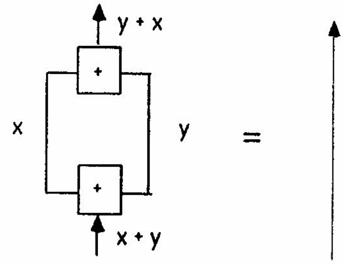
Most of the effort of the group was focussed on point 3, performing the algebraic simplifications. It was discovered that a number of simplifications could be performed similar to those performed in MACSYMA. The simplifications correspond to the laws of arithmetic or to algebraic identities. Some of the simplifications identified are shown in Figures 3 and 4.

Using identities such as these, it was possible to solve all six of the example problems submitted, using a disciplined hand simulation. It therefore appears that it would be feasible to perform the simplifications by computer.

An algorithm for the expression of a graph in bi-connected components (Sedgwick, 1983, pp 390-392), was found to be useful in identifying parts of the network as candidates for simplification.

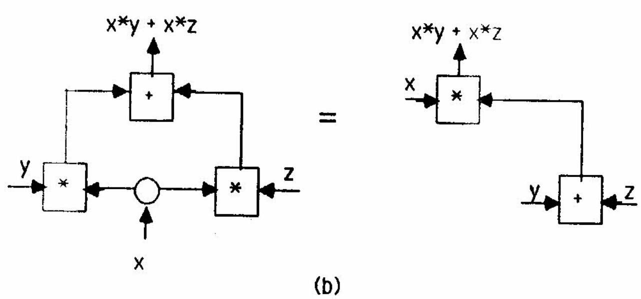
5. CONCLUSION

A plausibly programmable approach to the "blocked network" problem was found using the theory of algebraic simplification and of graph ismorphisms. The network representation of mathematical and logical systems seems to be a very convenient one for such problems.



A + B = B + A

(a)

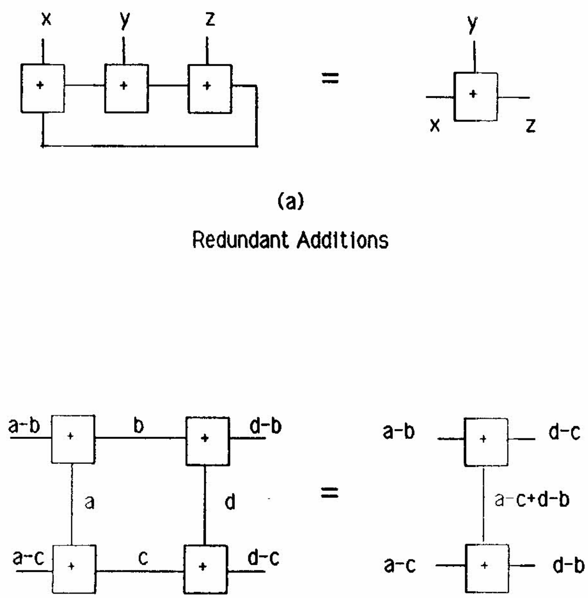


(b)

A * (B + C) = A * B + A * C

Figure 3

Simplification Using The Laws of Arithmetic



(a)

Redundant Additions

(b)

A More Complex Identity

Figure 4

Simplification Using Algebraic Identities

REFERENCES

Brander, J. and Royle, R. "ORION: A general purpose knowledge based problem solver" Proceedings of the First Pan Pacific Computer Conference, Australian Computer Society (1985), 925-943.

Sammut, C.A. and Sammut, R.A. "The implementation of UNSW-Prolog", Australian Computer Journal 15 (1983a), 58-64.

Sammut, R.A. and Sammut, C.A. "PROLOG, A tutorial introduction", Australian Computer Journal 15 (1983b), 42-57.