AN EXPERT SYSTEM FOR UNDERSTANDING EXPRESSIONS FROM ELECTRIC CIRCUIT ANALYSIS

Rostam Joobbani and Sarosh N. Talukdar

Design Research Center, Carnegie-Mellon University

Schenley Park, Pittsburgh, PA 15213

ABSLRACT

Circuit analysis is central to electrical engineering. It is important that students master the skills it requires. Most of cuaiit analysis is concerned with writing expressions that embody transtoamations, conservation laws and the electrical behaviors of various devices This paper describes work in progress or. an expert system that can examine the expressions that students write, diagnose the errors in them and suggest corrections. The expert system is implemented in OPS5 and uses the strategies of divide and conquer, forward chaining and hypoihesize and-test The system can now recognise most of the Common errors that students make in dealing with linear resistive circuits and operates at about the level of a competent human grader.

1 CIRCUIT ANALYSIS

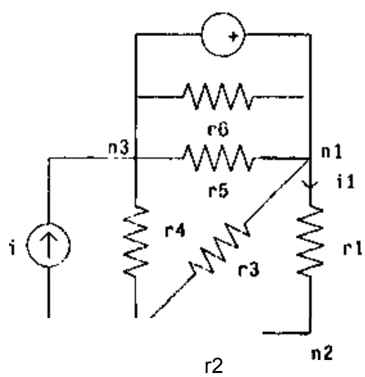
The general form of a circuit analysis problem is:

- given the configuration of a circuit and two subsets of its variables (one designated as known, the other as unknown);
- solve for the unknown variables in terms of the known variables.

By "configuration" we mean P specification of the type and location of each element in the uicuit. By "variable" we mean the value of a circuit element or a voltage or a current. An example of a circuit problem is shown in Figure 1.

The first stage in solving a circuit analysis problem is to assemble a set of equations relating the known and unknown variables. The second stage is to solve for the unknowns. Programs embodying numerical and symbolic procedures for the second stage are widely available. Therefore, in this paper we will address only the first stage.

Equations relating the known and unknown variables are generated by invoking Kirchhoff's conservation laws (the net current entering each and every node is zero; the net voltage around each and every loop is zero) and by using branch relations (that describe the electrical behavior of the elements in the circuit - for instance, voltage is equal to resistance multiplied by current in the case of a resistor). Many more



Figu re 1: An example: for the circuit above, express i 1 in terms of i, v and r 1, r6.

equations can be generated than is necessary. Therefore, part of the problem is to select a subset of all the possible equations such that the subset is complete and independent from the viewpoint of the unknown variables. Such subsets are not unique.

The task of assembling equations can often be made easier by network transformations that replace portions of the circuit with electrical equivalents that have fewer elements[1], as illustrated in Figure 2. Replacements can be represented by algebraic expressions of the form:

Replace $SC_1(r_1.....rm$ by $SC_2(s_1.....s_n)$

where SC denotes subconfiguration and the r's and s's denote the element values in subconfigurations.

In summary, the first stage of solving a circuit analysis problem consists largely of writing expressions that represent network transformations, conservation laws and branch relations.

1.1 Intellg Computer-Based Tutors

Circuit analysis is central to electrical engineering so it is important that students master the skills it requires. As with many other engineering and scientific topics, these skills are best learned through "doing", that is, through problem solving[2]. Problem solving sessions are most effective when they are

*This work was funded in part by the Desiijn Research Center of Carnegie-Mellon Univensty and by Found for Improvement of Pout Secondary education (FIPSE).

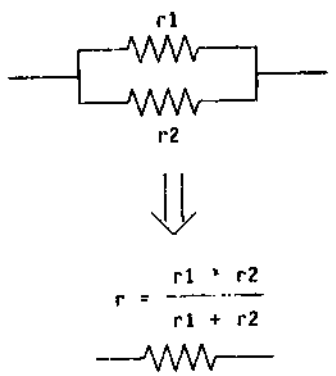


Figure 2: Replacing a pair of parallel resistors by a single resistor.

guided by a tutor that is able to understand what the student is actually doing, not just what he ought to be doing. We are in the process of building such a tutor. Among its components is an export system that understands the mathematical expressions that students write. The next section describes this expert further.

2 THE EXPRESSION-EXPERT

The purpose of this expert is to monitor each expression a student writes and quickly recognize any errors in it. (With beginning students we intend to print out and explain errors as soon as they occur. More advanced students will be allowed to complete attempts to solve entire prohle.ns before their errors are discussed.)

An advanced prototype of the expression expert has been written in OPS5[3] and Franz-Lisp[4]. A brief description of this expert follows.

The expression expert contains some basic mathematical knowledge, a fair amount of circuit analysis knowledge and an expanding store of student modelling/diagnostic knowledge.

The mathematical knowledge covers:

- the use of the operators: + , - , * , /, = and -> (replace)
- the transitivity law [a + b = b + a, a*b = b*a].
- the associativity law [a + (b + c) = (a + b) + c].
- equivalence [a + 0 = a, a*1 = a, a*0 = 0].

The circuit analysis knowledge covers:

- simple network transformations (series and parallel equivalents as well as the removal of shorted elements but not Y-Δ transformations and more complicated equivalents)
- Kirchhoff's current law
- branch relations for resistors, independent voltage sources and independent current sources

The student modelling/diagnostic knowledge has been collected from some of CMU's best teachers of circuit analysis and consists of a taxonomy of errors and the misconceptions most likely to cause them.

2.2 Approach

When presented with an expression, the expression-expert attempts to answer the following questions:

- What is the expression supposed to represent[7] (current conservation? If so, for which node? A branch relation? If so, for which branch? A network transformation? If so, of what sort?)
- Is the expression correct? If not, what are the errors and what misconceptions on the part of the student are likely to have caused them?

To answer these questions quickly and efficiently, the expert uses the strategies of divide-and conquer, forward chaining and hypothesizeand-test[5]. Four steps are involved:

1. Dissect the expression into subexpressions.
2. Determine what each subexpression is supposed to represent (a voltage, current or resistance). If any subexpression is not immediately recognizable, move on to the next subexpression.
3. From the recognizable subexpressions, identify the overall expression.
4. Generate and test hypotheses for the identities of the hitherto unrecognized subexpressions.

The following simple examples illustrate how the strategy works. Doth examples are based on the circuit of Figure 1.

In the first example the student enters a correct expression for current conservation at node n2. This expression is: (v2-v1)/r1 + (v2-v4)/r2 = 0. The expression expert proceeds as follows:

1. The expression is dissected into the subexpressions (v2-v1)/r1 and (v2-v4)/r2.
2. Each subexpression is examined separately. (v2-v1) is recognized as being the voltage difference between nodes n2 and n1. Then (v2-v1)/r1 is recognized as being a current flowing in ii. Similarly, (v2-v4)/r2 is recognized as being the current in tesistor r2.

3. Knowing that the expression is a sum of currents, the expert reasons that it represents a current conservations and searches for an appropriate node(s). This node is found to be n2. Now the expert checks the overall expression to see that it is correct (that there are no missing currents and all the signs are right).

As a second example, consider the current conservation equation for node n4. A correct equation is: **i + (v4-v3)/r4 + (v4-v1)/r3 + (v4-v2)/r2 = 0.** Suppose that a student makes the mistake of writing *n* instead of r3 in the equation. The expression expert proceeds as follows:

I. The expression is dissected into the subexpressions: i, **(v4-v3)/r4, (v4-v1)/r1,** and **(v4-v3)/r2.**

2. Using reasoning like that described in the previous example, the expert recognizes i, **(v4-v3)/r4** and **(v4-v3)/r2** as currents. However, **(v4-v1)/r1** does not make immediate sense and is left unrecognized.

3. The export ignores the unrecognized subexpression and concentrates on the others which, under examination, prove to be currents flowing out of a single node, namely, n4. Thus, of all the possibilities, the expression comes closest to being a current conservation equation for node n4.

4. In light of this information, it is clear that the unrecognized subexpression should represent the current through resistor r3, which it would if r1 were changed to r3. (One hypothesize-andtest cycle was required to achieve this)

Two points are worth noting:

- Hypothesize-and-test cycles are time consuming and are used only for unrecognized subexpressions, not for subexpressions that are recognized but out of context. For example, suppose that a student replaces the current in resistor r2 with the current in resistor r1. These two currents are the same. The expert recognizes this without having to employ any hypothesize-and-test cycles.

- To understand complicated mistakes the expert may have to use several hypothesize-and-test cycles. Suppose, for instance, the student writes: **i + (v4-v3)/r4 + (v4-v1)/r2 + (v4-v2)/r3 = 0.** The first two terms of this expression are currents that could apply to either node n3 or n4. The second two terms are mistakes. To examine all the possibilities takes several hypothesize-and-test cycles.

2,3 Status

The existing version of the expression-expert can correctly identify most of the common errors made by students. It is written in OPS5 and Franz-Lisp and uses MACSYMA[6] for solving systems of equations. It and its supporting structure, consists of 800 OPS5 rules, broken down as shown in Table I.

3 FUTURE WORK

Our plans for continuing work include expanding the expression expert's store of knowledge on student models and diagnostics, and extending its capabilities to include circuits with dependent sources This work is part of a project designed to produce intelligent tutors for basic topics in engineering and science . The first of these tutors is targeted for completion in 1986 when it will be deployed in a network of powerful work stations that will make it readily available to all the electrical engineering undergraduates at Carnegie-Mellon University.

Task/Function	No. of rules
Electrical engineering knowledge	291
Mathematical knowledge	14
Equation solving	53
Diagnostics/Tutoring	84
User interface	167
Student modeling	70
Hypothesize-and-test	50
Scheduling	14
Housekeeping	57
Total:	800

Table 1: A breakdown of the types of rules used.

References

1. Genturia, S.D., and B D. Wedlock, *Electronic Circuits and Applications,* John, Wiley & Sons, Inc., New York, pp. 55-60, 1975.

2. Anderson, J.R., C.F. Boyle, R. Farrell, and B.J. Reiser, "Cognitive Principles in the Design of Computer Tutors", *Internal paper, at the present time, Advanced Computer tutoring Project, Carnegie-Mellon U*niversity1984.

3. Forgy, C.L., *OPS5 User's Manual,* Department of Computer Science, Carnegie-Mellon University, July 1981.

4. Foderaro, J.K., *The FRANZ LISP Manual,* University of California at Berkeley 1981.

5. Barr, A., and E.A. Feigonbaum (Eds.), *The Handbook of Artificial Intelligence (vols I, 2, 3),* Los Altos, CA: Kaufmann, 1981, 1982.

6. *Macsyma Reference Manual,* The Mathlab Group, Laboratory for Computer Science, MIT, December 1978.

7. Joobbani, R., and S. N. Talukdar. "A Knowledge-Based Expert System for Tutoring in Electrical Engineering", *Proceedings, 8th International Computing Symposium* /CS-85Florence, Italy, March 1986.