ELSEVIER

Artificial Intelligence 111 (1999) 239–276

Artificial Intelligence

www.elsevier.com/locate/artint

Qualitative circuit models in failure analysis reasoning

M.H. Lee [1]

Department of Computer Science, University of Wales, Aberystwyth, SY23 3DB, Wales, UK

Received 9 July 1998

Abstract

The engineering task of failure analysis involves reasoning about the behaviour of a system using appropriate models of system components and structure. This paper describes methods of qualitatively modelling electrical circuits that support the requirements for certain combinatorially demanding forms of failure analysis. Minimal models, based on zero-order quiescent conditions, are examined and a particular formulation, known as CIRQ, is shown to be an effective and efficient model with strong intuitive features. Theoretical background is given and simulation algorithms are described. These models have been used as the basis for successful failure analysis software packages that solve large-scale real applications involving repeated behaviour inference. The contributions of this work include the development of minimal qualitative circuit models and simulation algorithms, an understanding of their relationship to certain graph-theoretic properties of circuits and the relevance of such models for fault modelling in FMEA tasks. The limitations of the approach are discussed and its relation to other work is examined.

Keywords: Qualitative modelling; Failure analysis reasoning; FMEA (Failure Mode Effects Analysis); Symbolic electrical circuit simulation

1. Introduction and motivation

Qualitative Reasoning (QR) is the branch of Artificial Intelligence research that concerns the modelling of physical systems and phenomena using imprecise or low resolution information [39]. A major goal for QR is to create tools that automate some of the engineering reasoning necessary in tasks such as design analysis, diagnosis, control, interpretation and exposition. This paper concerns the role of QR in modelling electrical

[1] Email: mhl@aber.ac.uk.

0004-3702/99/$ – see front matter
PII: S0004-3702(99)00032-6

circuits and describes the theory and background for our work on qualitative modelling of electrical circuits for supporting failure analysis and early design exploration. We examine options for minimal circuit representations and develop a steady-state model as a useful qualitative framework for a range of failure reasoning applications. We show why our particular qualitative formulation, known as CIRQ, has been very effective as the model that underpins several successful industrial failure analysis systems.

An important question in QR is "What is the minimum information that can support useful reasoning for a given problem class?" The meaning of "minimum" here refers to the use of abstractions and simplifications, which eliminate information by focussing on essentials (abstraction) and removing details (simplification). Abstraction can be seen as a process of approximation while simplification is a way of reducing resolution. However, this is a complex topic; see [18] and [33] for discussions of the different dimensions of information reduction in QR. The underlying premise of QR assumes that significantly reduced representations of variables or systems can capture their essence in a way that facilitates basic understanding and manipulation. The benefits of simplicity and abstraction are compromised by the lack of detail that accompanies coarse grain size, but we notice that rough approximations are very much in the spirit of preliminary design [12] and have a long tradition in engineering analysis [13].

Failure analysis is a vital task performed regularly and widely across all branches of engineering and failure analysis techniques are now an integral part of engineering design. The process of failure analysis involves inferring the global effects on a system caused by the existence of one or more localised faulty components, each of which may have a diverse range of failure modes. Such analysis is often performed at an early stage of design as a means of avoiding critical or hazardous situations and ensuring graceful degradation when components fail. Example failure analysis methodologies include FMEA (Failure Mode Effects Analysis), FTA (Fault Tree Analysis), PRA (Probabilistic Risk Analysis), HAZOP (Hazard Operability study) and SHERPA (Systematic Human Error Reduction and Prediction Approach). See [37] for an introduction to these methodologies.

Unlike diagnosis where faults are to be inferred from observed symptoms, in failure analysis the aim is to derive the consequences of known fault conditions. Thus any causal relations are usually less ambiguous to interpret and the nature of the required reasoning is often deductive rather than abductive. However, like most other engineering reasoning tasks, effective failure reasoning requires considerable expertise and extensive knowledge of the domain. Practising engineers have a deep intuitive understanding of how artifacts work and employ this to guide and direct much of their thinking [3]. Their expertise includes knowledge of consequences, strong preferences, use of patterns and clichés, and common-sense assumptions. Despite the power and mathematical sophistication of existing engineering analysis software tools, human designers and analysts seem to exploit simple intuitive models in a range of circuit reasoning abilities. For these reasons, and notwithstanding the existence of prescriptive and well established procedures, only recently have failure analysis methods been successfully automated.

We have investigated some extremely restricted representations of electrical circuit properties because we wish to explore the value of minimal models and also because some failure analysis methods demand very efficient algorithms for their repetitive processing requirements. Our approach is to explore the most minimal representations first and

examine their interpretations against the criteria for satisfying some basic circuit reasoning tasks. We explore these ideas in the context of an electrical failure analysis technique but the methods extend to other safety assessment tasks and related areas such as design exploration, design modification and design diagnosis.

This paper is structured as follows. The next section describes the task characteristics and requirements of failure analysis. Section 3 then highlights the difficulties with numerical methods. We describe our theory for electrical system modelling in Section 4 and then Section 5 outlines the CIRQ circuit simulation algorithm. Section 6 gives a worked example case study for both numerical and qualitative solutions thus illustrating their differences. Section 7 examines the extent to which the method satisfies the original requirements and Section 8 examines alternative circuit algebras. The final three sections describe the limitations of the technique, review related work and summarize what has been achieved.

2. Requirements for an electrical circuit failure analysis tool

The requirements for failure analysis of electrical systems have similar core features for many of the assessment methodologies. We chose to examine the task requirements of FMEA (Failure Mode Effects Analysis) as this is an important and very widely used technique. First we describe the idea of electrical FMEA, then specify the key characteristics of the task in general and then illustrate with specific requirements from our experience.

2.1. The nature of the FMEA task

In applications with complex physical wiring systems, e.g., avionic and automotive systems, manufacturers are very concerned about the consequences of any potential fault in the wiring. The integrity of major system functions can be threatened by relatively minor electrical defects. FMEA is the industry-wide standard methodology for analyzing the effects, usually in terms of safety, of all variations of a class of possible faults. As FMEA documents give a formal evaluation of the seriousness of any safety critical behaviour, their production is now an increasingly important part of the design process for many complex engineered products. Legislation in many countries now requires FMEA work to be performed extensively on new product designs.

The way FMEA is carried out is by proposing and answering a comprehensive series of “what if this happens” questions. The design engineer identifies a set of components and then, for each component, lists the different ways in which it could fail—this gives a set of component faults. Then the engineer considers how the presence of each individual fault affects the other parts of the circuit—the system changes due to a component fault define a failure mode of the system. The next stage is to estimate the implications and severity of each failure mode on the overall system. In this way, the global effects of each possible fault and all failure modes are examined and recorded. In order to understand the implications of a system failure it is necessary to appreciate the role that it plays in the total application. This requires considerable non-electrical domain knowledge, for example, loss

of power to a light bulb will have quite different implications depending upon its function. The bulb might act as an indicator to the driver or it might illuminate the road ahead. Thus knowledge of the *function* of the system or subsystems is very important. In order to record the severity of a failure, standard methods for scoring the criticality of effects have been devised and include factors for likelihood of failure, ease of detection and seriousness of the outcome. The final FMEA report contains such severity ratings for all failure modes.

In nearly all electrical FMEA work the class of faults is limited to various forms of short circuit and open circuit. Even in more subtle failures it is often deemed satisfactory to represent the change to the system as an open or short circuit event. As this is true for complex applications such as aeroplanes and automobiles we will adopt the same fault classes.

By its exhaustive nature FMEA is a very tedious process and yet demands the expertise of a professional engineer. Until very recently the only tools available to assist in the process were bookkeeping programs that help with some of the clerical aspects. Using the methods described in this paper a world-first industrially deployed software system has been produced that can perform routine reasoning to determine the extent of electrical changes in a circuit due to postulated faults [27]. While the judgement and domain knowledge of the skilled engineer are still involved in controlling the fault schedule and assessing the risks of different system failures, the FMEA process has been speeded up by several orders of magnitude (from weeks to hours).

2.2. The task requirements

Electrical and electronic systems display a very modular structure, being strongly based on the notion of a component. Components are specified with an internal structure and/or function and have a defined interface to the external world. Usually the internal structure is fixed but parameterised and components are created by instantiating from selected archetypes. Network topologies are then built up by connecting components together by giving assignments to their ports or terminals. This modularity is very important for engineers, possibly because it assists analysis and intuition, and so we must aim to preserve this inherent feature in any qualitative representation.

Such circuits fall naturally into the category of *device* ontologies [11], where systems are composed of sets of individual components connected together by specified interconnections between component ports. System behaviour is then inferred from the (known) components' behaviour mediated by the network connections to other components. With such strongly modular system structure it is common practice to maintain a catalogue of component types, often organised as a hierarchy of classes with inheritance of properties and defaults. We will assume that all physical entities in the electrical world are defined as components, ranging from simple wires to complex subsystems. Examples of two terminal devices include wires, resistors, fuses and motors. Components with more than two terminals include connectors, switches, relays and complex component boxes which encapsulate specific proprietary circuits. We also assume that all complex components can be defined in terms of an internal circuit that is composed of other circuit elements.

We now consider some key features of this problem domain:

Circuit based. The base system is a *circuit* consisting of multi-terminal components connected via their terminals. Components are modules that may be represented as parameterised instantiations of prespecified types drawn from a component catalogue. The catalogue entries define the number of terminals for each component and how they are connected internally. Note that *all* physical entities are components; this includes connecting wires.

Hierarchical decomposition of components. Electrical circuit components are defined in terms of a set of labelled terminals and an internal structure. A *base set* of primitive components is prespecified without reference to any other components and consist of the fundamental circuit elements, e.g., wires, resistors, switches, etc. Other components have internal circuits that are specified as a set of connections between components from the base set or (non-recursively) from the non-base components. This essentially hierarchical structure means that all circuits are reducible to a collection of only base components.

Fault classes. Faults in a circuit may involve connectivity errors or component faults. The classes of connectivity fault commonly used in electrical FMEA are open circuits and short circuits. Component faults may be defined as abnormal modifications to a components internal structure. These may involve changes to a component's electrical parameters or topological changes to the internal circuit configuration.

Effects. The effects of interest that are to be recorded in a fault scenario will vary depending upon the application requirements. At the most detailed level this will cover all electrical variables that have changed as a result of the fault. Less demanding levels involve recording changes in just one variable, e.g., current, or in just one specified sub-area of the system. The least detailed but quite effective information is the identification of those components that have changed their activity state in some particular way.

Large scale. Although most components tend to have relatively simple internal structure, some may contain very large circuits. In addition, applications can have arbitrarily large circuits and component counts of thousands are not uncommon. This means that any processing methods must scale up to deal with large numbers of components with acceptable time and space complexity.

Forward analysis. The purpose of failure analysis is to deduce the direct effects caused by explicit changes (faults) from a normal mode. There are no requirements for explanation or causal reasoning and, unlike diagnosis, there is no ambiguity about the cause of an effect.

Static analysis. FMEA analyses take place on a static snapshot of the system. Unlike diagnosis, all faults are treated as hard (even intermittent cases) and so there is no requirement for dynamic analysis. State transitions are of interest but these can be treated by repeated analysis over sequences of static snapshots.

Exhaustive. As no failure case must be missed, *all* specified faults in the fault classes must be applied to *all* relevant components in the system. Thus the task is very repetitive and must be efficient for repeated analysis.

Multiple faults. Advanced FMEA systems may require the effects of several simultaneous faults to be analyzed. However, this appears to be rarely performed in practice.

Multiple domains. Although the faults are to originate within an electrical system, the effects of the failed system behaviour will have consequences outside the electrical domain. These effects on other systems or on the non-electrical aspects of an application must be considered in FMEA. This involves much engineering experience particularly including knowledge of the function of the sub-systems. See [25] for details of the use of function labels which capture such implications in our FMEA system. We do not discuss this aspect of FMEA here.

Criticality. The effects of a failure must be assessed in terms such as seriousness and likelihood. However, this involves additional non-electrical domain knowledge that is either precompiled or known by engineers. Component reliability data, safety requirements and functional knowledge are all involved in assessing the criticality of failure cases. This post-processing stage is excluded from discussion of our system but see [26] for further information on the automation of the effects evaluation stage.

2.3. A specific example task

From our work with industrial engineers we now describe the requirements for a typical FMEA process for automobile wiring systems. This gives a concrete illustration of real requirements. The electrical FMEA task has the following features:

Circuit structure. An input circuit is defined as a graph, $C(\mathit{components}, \mathit{connections})$, where *components* specifies a set of devices from a component library and *connections* is a net-list of pairs defining associations between (the labelled) component terminals. Components may have associated parameter values. The internal networks of the components can be accessed from the library definitions in order to translate the whole circuit C into a graph, E, consisting of T *nodes* and R *weighted edges*:

$$C(\mathit{components}, \mathit{connections}) \rightarrow E(T, R)$$

The nodes include the terminals where voltage but not current can be assigned, and the edges are the conducting flow paths that can be assigned current but not voltage values.

Normal modes. Assuming that the circuit contains m switches or other elements that can significantly change their internal connectivity characteristics, [2] then there will exist up to $2^m = n$ normal mode states: $S_1, S_2, \ldots, S_n$. For each normal mode circuit state we can apply a procedure, **Solve**, to find all current values in the edges, all voltages at the terminals, or any selection of these, i.e.,

$$\mathbf{Solve}\big(E(T, R)\big) \rightarrow S_i\,.$$

[2] We assume binary switches here; other devices or switches may have more than 2 configurations but their combinations can be mapped into states in the same way.

For this application we wish to know, for each normal mode state, which components are electrically active and which are inactive. We define an active edge as one whose nodes each have a path (of less than infinite resistance) to different supply terminals, such that the paths are disjoint and contain no nodes that link them by any zero resistance paths. This means an active edge cannot be part of a branch that is either shorted out or open-circuit.

Fault classes. The most commonly used fault schedule in electrical FMEA is to apply an open circuit to each conductor and short circuits between each terminal and each supply potential. General short circuit faults between all terminal pairs are not usually considered due to their enormous combinatorial complexity and an implicit assumption that their worst effects will be covered in the supply short circuit analysis.

Fault application. The above faults translate into the graph E as (a) open-circuits applied to each edge R, and (b) short-circuits, first to one supply terminal and then to the other, for each node T. We can view these as changes of resistance applied to the original circuit. Thus, open-circuit is a change of edge resistance to ∞ and a short-circuit is a new path of resistance value 0 between the node and supply. Notice that this covers component faults as well as circuit faults, as the graph E contains the internal networks of all the components. We define Δr as the *fault change list* that specifies individual fault changes and **Modify** is a function that applies the changes to a given circuit graph. Notice that faults are independent and any number can be specified in a given Δr. However, although our system allows multiple faults in this way, the normal practice is to take each fault in turn and so Δr will contain only one fault change in what follows.

Failure modes. The result of applying a fault is that some components may experience a change in current: $\delta I \neq 0$. Each instantiated fault, k, will produce a failure mode state S_i^k corresponding to each normal mode state S_i. It may be possible to reduce the number of distinct failure cases by recognising that several faults are exactly equivalent. For example, an open circuit fault need only be performed once on a set of edges connected in series.

Effects. The FMEA report must record the effects of each fault by comparing normal and failure modes. We solve the circuit for S_i^k and then find the difference from the normal mode to produce a component change list:

$$\mathbf{Diff}\left(S_i, S_i^k\right) \rightarrow (P_a, P_d)_i^k$$

where P_a is a list of all components that have become active and P_d is a list of all components that have been deactivated, as a result of fault k on normal mode state i. If the component change list is empty then the given fault has had no effect. We notice that general current changes, δI, are of no interest; only transitions between active and inactive are recorded.

From these requirements we can see how the task must be performed. The basic process is as follows:

For each normal mode, i:
Solve$(E(T, R)) \rightarrow S_i$
For each fault class:
For each edge (or node, depending on fault class) $\in E$:
Create fault case Δr^k
Solve(**Modify**$(\Delta r^k, E(T, R))) \rightarrow S_i^k$
Diff$(S_i, S_i^k) \rightarrow (P_a, P_d)_i^k$

The combinatorial variations of the circuit size, the fault classes and any switched components can reach significant levels. An example calculation shows this effect. Let there be 3 fault classes (say 1 for edges and 2 for nodes, as above), then in a circuit of $R = 50$, $T = 35$ there will be 120 potential fault sites. If there are 5 switches, then there are 32 normal mode states and thus there are 3840 potential failure modes. Hence even quite small circuits can require many applications of the solution method.

This raises further considerations about the task:

Efficiency. From above we see that the effects of circuit changes must be deduced very rapidly as many iterations must be performed in order to satisfy the exhaustiveness requirement.

Intelligibility. The method used must be comprehensible to engineers who will want to verify the analysis process by tracing arbitrary fault trails. Although this is a difficult requirement to assess it will be enhanced if the circuit representations have some clear intuitive appeal.

Static analysis. There are no requirements for dynamic effects to be analyzed; transients, glitches and time varying features are not included in basic FMEA. We can thus always assume quiescent current conditions have been reached.

Complex components with internal state. Programmable devices, e.g., PLCs, CPUs and other complex electronic devices can be defined in terms of their internal logic and the respective internal circuits corresponding to different internal states. This is achieved with tables that define the internal network configuration between the terminals for each distinct component state. A simple example is a relay, where different patterns of connectivity between the terminals are given for the energised and unenergised states. For example, if I_R is the current through the relay's coil and SW is its contact switch, then a state table entry might record (see Table 1).

Higher level analysis through multiple state transitions. If the circuit contains any components with state then it may be necessary to run a series of analysis steps to deduce the

Table 1
Component state table

Condition	State
$I_R = 0$	$SW = open$
$I_R = active$	$SW = closed$

outcome of a behaviour mode. Using the relay example, if the first analysis shows a relay to be energised then its switch contacts change and the circuit enters a new state. This requires another analysis to find the circuit elements that are affected and the process repeats until steady-state is reached or cycles are detected. This applies to both normal and failure modes and, in general, we should replace our states, S_i, with vectors of states, $\bar{S}_i$. For simplicity we will ignore this repeating process but in practice we would replace **Solve** in the solution process above with:

Solve$(E(T, R)) \rightarrow \bar{S}_i$
Scan all state tables, create Δr
While $\Delta r \neq 0$
    $\bar{S}_i$ + **Solve**(**Modify**$(\Delta r, E(T, R))) \rightarrow \bar{S}_i$
    Scan all state tables, create new Δr

3. Problems with quantitative methods

The traditional approach to circuit analysis is to employ a numerical analysis package such as Spice [38] or Saber [28]. For any connected network of linear elements a set of simultaneous equations, known as mesh or node equations according to whether voltage or current is the primary variable, can be generated. These circuit equation solvers convert the topology of the circuit from a specification either in graphical form or as a list of connections (the net-list) into matrix representations for efficient matrix processing. The particular approach used in Spice, for example, is the branch-admittance matrix.

Thus, such packages are numerical versions of **Solve** and can produce accurate values of the voltages and currents at all points in a circuit:

$$\textbf{Solve}\big(E(T, R)\big) \rightarrow W_i$$

where W_i is a numerical solution for the circuit graph E. The circuit equations must be solved both with and without a proposed fault and the effects are found through a comparison of the results to produce the component change list:

$$\textbf{Diff}\big(W_i, W_i^k\big) \rightarrow (P_a, P_d)_i^k$$

However, although **Solve** poses no difficulties, **Diff** is much more problematic for numeric data. The determination of what constitutes a *significant* change in a real numbered system is subtle and not straight-forward. Considering our circuits as connected resistive meshes with one or more complete flow paths (excepting entirely separate branches joined only at the supply terminals) then we see from standard circuit theory, e.g., the compensation theorem, that any injection of current or any component value change will perturb *all* the flow values in the system *to some extent*. Because we must distinguish between the many trivial perturbations and the ones which represent failure effects we need to define some form of numeric criteria that can be used by **Diff** in order to decide on *significant* changes. However, any such change thresholds or tolerance rules will vary across the circuit, being potentially different both for each component *and* for each normal mode. For example, in our automotive application domain, the voltage drops along some

of the conductors in an automobile are important and must be closely monitored while others are allowed to vary with little restriction. Similarly, the definition of zero current is not a simple numeric matter but requires a tolerance criteria below which any current flow is considered to be negligible.

Consequently **Diff** will need to be programmed with large numbers of change detection criteria and this reflects the difficulty in interpreting the results from **Solve** when real numbers are being used. The burden of work shifts from numeric equation solving to the domain dependent interpretation of all the electrical changes in the output. In other words, the numerical model in the analysis system is not very appropriate for the task and this leaves much human effort required to interpret the results.

Another problem with numeric analysis packages is that it is much easier to attempt to optimize circuit parameters rather than circuit topology. Various simulation schedules may be applied to parameters to evaluate sensitivity and performance but if a circuit requires a configuration change to match some requirements there are few guidelines or methods to assist any automated process. This is the case in FMEA where the fault modes include many topological changes.

Our reservations are confirmed by reports from ECAD experts who have access to considerable data on engineers' experience, e.g.:

> "Graphical interfaces can present simulation data in more convenient forms, but they do not interpret their meaning nor do they reduce the number of simulations required in order to gain a feel about some circuit performance. It has thus become clear that numerical descriptions of circuit behaviour do not convey understanding about the operation of a circuit. In fact, the use of such numerical-only systems can impose a barrier to the development of insight." [19]

Of course, numeric detail is often necessary to resolve failure cases but we pursue the QR approach here as a complementary aspect of FMEA. We note that engineers are often able to infer failure consequences with coarse levels of detail and it is interesting and valid to explore QR as a first-cut analysis mechanism before any finer scale analysis.

4. An electrical ontology based on resistance

We adopt the conventional working approximation that any circuit can be represented by a suitable network of interconnected lumped parameter elements. In this work, we consider resistance as the *only* primitive electrical property in the ontology. Thus any system to be modelled and analyzed must be represented in terms of some configuration of resistances. This might seem restrictive but the requirements only specify static analysis in the steady state. We therefore assume quiescent current conditions and are prepared to ignore first order or higher effects. However, FMEA does include sequential state analysis and sequential events involving identifiable state changes should be covered.

This approach views resistance as a first approximation model of any energy absorbing electrical component. We can thus model motors, relays, connectors and various forms of load but not energy storage devices like inductors or capacitors. Extensions to handle capacitance, inductance and sinusoidal steady-state circuits are discussed in Section 9.

4.1. Qualitative resistance

A qualitative algebra consists of a representation scheme for variables, often known as the *quantity space*, and a set of operators for representing relationships or performing transformations. As resistance is a positive valued quantity [3] any qualitative representation must map onto the positive reals, $\mathcal{R}$. It might be supposed that the minimal representation of a resistive mesh will use a two-valued edge weight that corresponds to presence or absence of resistance. However, consideration of the FMEA task shows that we must be able to distinguish components that are inactive because either (a) they are bypassed by a very low resistance path or (b) they are supplied by a very high resistance path. The first case (a short-circuit) demands that we distinguish a conductor of effectively zero resistance from an energy absorbing load resistance and hence, to model shorted out components, we need (at least) two different levels of resistance that pass current. The quantity space for this can be given as $[0, +]$. In the second case, we need an additional symbol, ∞, in order to represent an open-circuit component. It might appear that an infinite edge value is unnecessary as the same effect can be achieved by disconnecting parts of the circuit topology. However, the requirement to model open-circuit components (such as switches) is best satisfied by using edge weights as parameters. Such parameters provide a model that can change in accordance with system operation and conveniently localise and identify the point of change.

It is easily shown that all the two-valued combinations of $[0, +, \infty]$ are inadequate for our purpose. Thus we conclude that a quantity space with three values $[0, +, \infty]$ is the minimum necessary for the FMEA task. This space can be viewed as limit points at 0 and ∞ and the intervening open interval $(0, \infty)$ and can be interpreted as near perfect conductors, insulators and energy absorbing devices, respectively. This has strong correspondence with engineers' intuitive notions of electrical conduits.

4.2. Qualitative resistance transforms

There are various methods for combining resistances in order to simplify the solution process. Graph-theoretic methods perform graph reduction and transformation by using rewriting or replacement rules [1]. For electrical circuits the most well known are the series and parallel reduction rules and the star/delta transform equations. Some qualitative work [21] uses all these transforms to reduce a circuit down to its equivalent single resistance value in order to solve and then, by reverse composition, assign current values. We find this unnecessary and use a direct traversal method that solves all circuit configurations but applies only the series/parallel rules.

The physics of series and parallel circuit reduction are satisfied by numerically summing their resistances or their conductances, respectively. A qualitative version of this physics can use operations on ordinal relationships if we rely on qualitative resistance being ordered: $0 < + < \infty$. Then the sum of a set of qualitative resistances in series is given by the largest value. Similarly, a collection in parallel gives an equivalent which takes the

[3] We ignore the concept of negative resistance that can occur in certain active systems.

Table 2
The qualitative reduction rules Max R and Min R

A	B	A & B in series	A & B in parallel
∞	∞	∞	∞
∞	$+$	∞	$+$
∞	0	∞	0
$+$	∞	∞	$+$
$+$	$+$	$+$	$+$
$+$	0	$+$	0
0	∞	∞	0
0	$+$	$+$	0
0	0	0	0

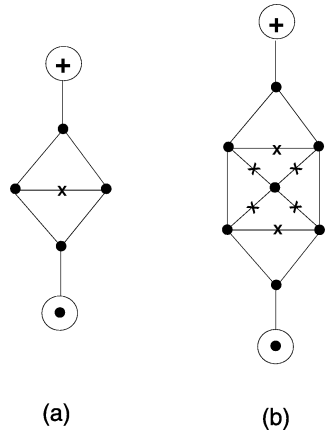

Fig. 1. Examples of non-SP reducible circuits.

minimum value. These Max and Min operators are shown in Table 2 for our three-valued quantity space.

Circuits that can be reduced to a single equivalent resistance by repeated application of the series/parallel rules are called SP reducible. Many circuits have a topology that is not SP reducible, some examples are shown in Fig. 1. For numerical solution, any circuits that are not SP reducible require the application of additional transform rules such as the star/delta conversion equations. However, for qualitative solution we show that our method can analyse *any* topology of circuit, even non-SP reducible circuits, using only the series/parallel rules. This will be explained in the context of the algorithm in Section 5.1 and further discussed in Section 10.

We now introduce a small but very valuable extension. We allow any *positive integer*, $n \in \mathcal{N}$, to stand for the qualitative resistance value $+$. Thus the ordering $0 < + < \infty$ is

Table 3
Series/parallel circuit reduction using Sum R and Min R

A	B	A & B in series	A & B in parallel
∞	∞	∞	∞
∞	m	∞	m
∞	0	∞	0
n	∞	∞	n
n	m	$n+m$	$\min(n,m)$
n	0	n	0
0	∞	∞	0
0	m	m	0
0	0	0	0

maintained because $0 < \mathcal{N} < \infty$ and any edge with an integer label [4] is considered to be a resistive load, exactly equivalent to the previous label $+$. Now two branches with positive integer labels, n and m, can be combined into a single equivalent by using any suitable operation that maps their values into another positive integer. Functions which satisfy this condition must also be symmetric and include: Sum, Max, Min, etc. [5] We have found that replacing Max with integer summation in series circuit reduction produces a very useful labelling scheme. Table 3 shows this modification.

Any integer can now be used in component definitions for an edge value but for the present models we restrict all resistive loads to unity. We allow the symbol ℓ to represent a load of value unity and so the set of qualitative component resistance values is now $[0, \ell, \infty]$. Applying the above reduction rules to a mixed series and parallel circuit with unity loads will return a value that represents the minimum path length between the circuit terminals *in terms of the number of load resistance edges*.

4.3. Qualitative voltage and current

Voltage and current are the only variables associated with a quiescent resistive mesh. They have different measurement characteristics and are complementary in their physical interpretation. Current is a flow (or "through") variable and therefore has direction and can be measured at any point on a flow path. However, in the present specification we do not require directional information, it is sufficient to know if a component is active or not. [6] We defined an inactive edge as one that is part of a circuit that is either shorted out or

[4] It is obvious that our notation for the natural numbers excludes zero.

[5] Such functions can produce neither zero nor ∞ from a finite number of operations on non-zero positive integers.

[6] In fact, directional information can be useful in some FMEA. We discuss the extent to which this may be recovered in Section 9.

open-circuit and an active edge as having two disjoint paths to the supply terminals. These can be represented by two values $[0, +]$ to give the qualitative space for current.

Voltage is a potential (or "across") variable and is measured at one point relative to another (reference) point. The minimal representation for qualitative voltage is also two-valued, corresponding to the two terminals of a supply or source and we use the qualitative values $[0, +]$ for voltages at the supply terminals and the symbols $\odot$ and $\oplus$ to represent the actual supply terminal nodes in the circuit graph. However, with 0 as the supply polarity for ground it could be impossible to distinguish a connection to ground from one to a disconnected sub-circuit. We assume voltage will be measured with a voltmeter that can detect continuity and assign the symbol $\emptyset$ for the voltage of a point with no resistive connection path to either supply terminal. It is important to note that $\emptyset$ is a special symbol implying a "floating" or disconnected section of circuit and is *not* a voltage level. Although not strictly necessary, we also introduce another symbol, $\sim$ to indicate the voltage of a node that is connected to an active branch between the supply potentials. This involves no computational cost and proves very useful. Again, it is important to state that both $\sim$ and $\emptyset$ are not strictly necessary and are not used as values in the circuit algebra that follows. They could be removed by the replacement rules: $(\sim \longrightarrow +)$ and $(\emptyset \longrightarrow 0)$.

To summarise, our voltage quantity space is $[0, +]$ but our symbol set is $\{0, +, \sim, \emptyset\}$.[7] Parity with one of the power terminals is signified by $+$ or 0; this can occur either through a direct connection to a supply terminal or in an inactive branch where all nodes are at the supply level. The symbol $\emptyset$ indicates an inactive node with no connection path to either supply terminals. The symbol $\sim$ can be assigned to an active node when its voltage is neither of the supply potentials.

4.4. Qualitative Ohm's law

For the FMEA task we need to find the currents (or voltages) at various points in a circuit given some changes in component resistance or local wiring modifications. There are many circuit theorems that could be used including those of Thevenin, Norton, Millman and the reciprocity theorem. However, we find the basic network tools of Ohm's law, Kirchoff's loop and junction rules and the series and parallel equivalent resistance transforms are sufficient for this purpose.

Considering the constraint imposed by Ohm's law ($V = I * R$) on the above value sets, we can deduce the voltage difference for any given qualitative current through a resistance as in assignment Table 4.

There are two problems in the table, which we identify by using Morgan's nomenclature [4]. The third line gives an ambiguous result because any voltage may exist across an insulator without current flow and the last line is an impossible case, indicated by the symbol $\sqcup$. Although qualitative multiplication is well defined (unlike addition) and therefore returns valid results for $V = I * R$, infinite voltage is an unrealistic situation in the physics of our resistive world. Infinite resistance has physical meaning as an insulator and we take it as axiomatic that such components never pass any current. However, physical limits on

[7] In previous papers we have used $\{-, +, \sim, 0\}$ rather than $\{0, +, \sim, \emptyset\}$. This notation has some advantages but can cause confusion if 0 is wrongly interpreted as lying *between* $-$ and $+$.

Table 4
Qualitative voltage for given current and resistance values

Current	Resistance	Voltage
0	0	0
0	+	0
0	∞	?
+	0	0
+	+	+
+	∞	⊔

Table 5
Qualitative current for given voltage and resistance values

Voltage	Resistance	Current
0	0	?
0	+	0
0	∞	0
+	0	⊔
+	+	+
+	∞	0

Table 6
Qualitative current for 5-valued resistance with 3-valued voltage

Voltage	Resistance	Current
0	0	0
0	0	+
0	0	*max*
0	*lo*	0
+	*lo*	+
+	*lo*	*max*
0	+	0
+	+	+
max	+	*max*
0	*hi*	0
+	*hi*	+
max	*hi*	+
0	∞	0
+	∞	0
max	∞	0

voltage and current are reflected in cases that are either ambiguous or cannot occur in practice. There is also another inherent ambiguity in that the first and fourth lines in the table give the same result; this is reflected in the first line of the complementary table for current, Table 5.

We see the limitations in this formulation: if $R = \infty$ then we cannot deduce V from I and if $R = 0$ then we cannot deduce I from V. There are thus only four useful assignments in the tables for Ohm's law.

A finer definition of qualitative voltage and current could employ three-valued variables for positive quantities, for example, $[0, +, \max]$.[8] (We prefer max rather than ∞ as the limit on voltage or current to indicate that these are not ideal sources in many systems.) Analysis shows that this only gains three more entries in the table for Ohm's law and offers very little additional benefit for the extra resolution. Another option is to increase the resolution of resistance, e.g., use a 5-valued resistance variable as shown in Table 6. However, examination of a range of variations such as these does not expose any with significant advantages over our minimal models. Although there is reason to believe that engineers sometimes use more populated quantity spaces, the higher valued versions only realise a few of their combinatorial possibilities as valid physical interpretations because many collapse into ambiguous cases. Consequently, they only offer a marginal increase in descriptive power over the two and three valued versions. Also operations on discrete n valued variables rapidly gain in complexity with increasing n. For these reasons we continue to pursue our minimal approach.

5. A qualitative circuit solving algorithm: CIRQ

We now describe our qualitative version of:

$$\textbf{Solve}\big(E(T, R)\big) \rightarrow S_i$$

Given a resistive network and a single constant voltage power source we need to find if current is flowing in each edge of the circuit graph and so the output of **Solve** should be the list of edges, each labelled with either 0 or $+$. Our qualitative **Solve** algorithm has two separate stages: first all nodes are labelled with path resistance parameters and their corresponding qualitative voltages, then the second stage identifies the active flow-paths by labelling the edges. The notion of path resistance is an important feature of our method. We define the *path resistance* $PR(s, t)$ between two nodes, s and t, as the value of a single equivalent resistance found by applying reduction transforms to the sub-networks that comprise all paths from s to t.

5.1. Phase 1: Node resistance labelling

The first stage algorithm labels all nodes in the network with their path resistances from both supply terminals. Each node is assigned a pair of variables for storing these *forward* and *reverse* path resistances, known as f/r. The convention is adopted that forward resistance gives the path value to the $\oplus$ node and reverse gives the path value to the $\odot$ node. Thus, for node n, $f(n) = PR(n, \oplus)$ and $r(n) = PR(n, \odot)$. We also define the *node resistance* of a node, $NR(n)$, as the sum of its path resistances, $NR(n) = f(n) + r(n)$.

The labelling algorithm, which is based on a version of Dijkstra's shortest distance algorithm [8], begins from $\oplus$ and assigns to each node's forward label the minimum path resistance value from the supply. Then the reverse labels are assigned by a repeat process starting from the other supply terminal. All f/r values are initialised to ∞/∞, so that any

[8] This should not be confused with using $\{0, \sim, +\}$ for a 2-valued quantity space, as discussed in Section 4.3.

disconnected sub-circuits that cannot be reached from the supply terminals are correctly labelled on exit.

procedure ***label-path-resistances*** (start, circuit-graph, direction)
if direction = forward then r-path := f
else r-path := r
let U := the set of all nodes (T) in circuit-graph (E)
let r-path(start) := 0
forall nodes $v \in U$, $v \neq$ start, let r-path(v) := ∞
repeat
find a node $v \in U$ for which r-path(v) is a minimum
for each node $w \in$ adjacent(v) with edge resistance R do
if $w \in U$ then
r-path(w) := **parallel**(r-path(w), **series**(r-path(v), R))
if $R = 0$ then add w to supernode V
end-foreach
remove v from U
end-repeat
end-label-path-resistances

The algorithm operates by exploring adjacent edges from the lowest valued nodes. The functions **parallel** and **series** apply the parallel and series equivalent resistance transforms from Table 3. As a new edge from v to w is explored its resistance, R, is added to the parent node value, r-path(v), using the series aggregation rule and the result is then combined with any existing value representing a parallel path, r-path(w), using the parallel aggregation rule. Fig. 2 illustrates this process. In Fig. 2(a) all edges have resistance value ℓ and the node u has produced labels for its successors w and v. Node v is currently being processed

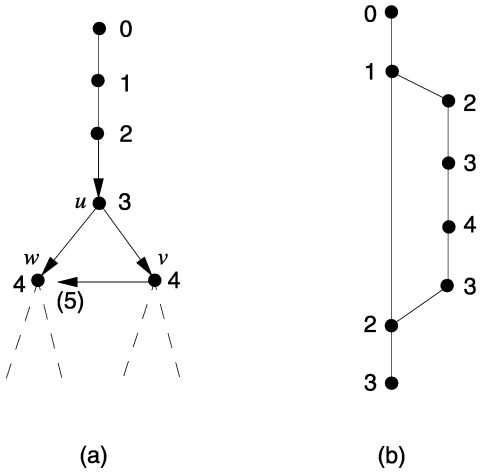

Fig. 2. The series/parallel labelling process.

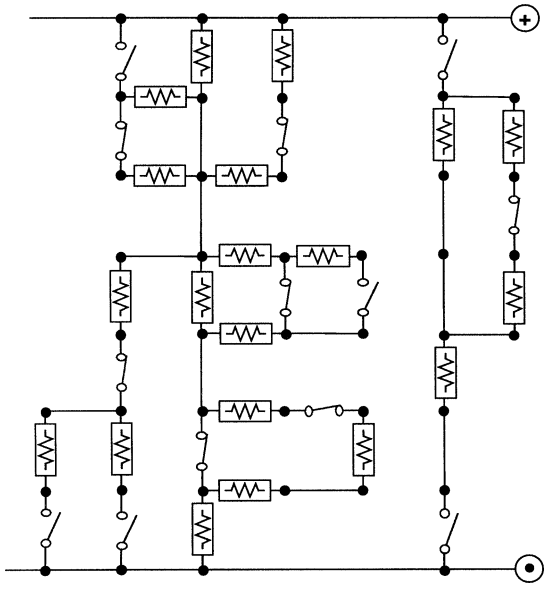

Fig. 3. An example circuit.

and so $4+1$ is to be combined with the existing value of 4 at node w. Thus a final result of 4 is produced:

$$\begin{aligned}\text{r-path}(w) &= \textbf{parallel}\big(\text{r-path}(w),\ \textbf{series}\big(\text{r-path}(v),\ R\big)\big)\\ &= \textbf{min}\big(4, \textbf{sum}(4,1)\big) = 4\end{aligned}$$

This process labels all nodes with their minimum path values to the designated supply node. Clearly, any quantitative form of this traversal process would fail because combining the branches of a parallel circuit would affect the previously calculated values on contributing series sections. For example, completing the parallel circuit at w would alter the value previously computed for v. It might be expected that this problem would also occur in qualitative cases where one series branch has a higher value than another. However, this does not happen because of the order followed by the labelling process. Fig. 2(b) shows how a long series branch is labelled. This topology causes the algorithm to enter the right-hand branch from both ends and thus each node is assigned its correct minimum path length from the start node. It is this combination of the series/parallel reduction rules with the traversal scheme that accounts for the success of the method—the equivalent resistance requirements have been converted into graph-theoretic properties (minimum path length). This explains why the method works for our qualitative formulation and also why this algorithm can label all topologies, even non-SP reducible circuits.

A novel feature of the algorithm is the generation of supernodes.[9] A *supernode* is a set of circuit nodes that are connected to each other by paths of resistance $R = 0$. All

[9] This concept has similarities with hyperedges in graph theory but is not identical.

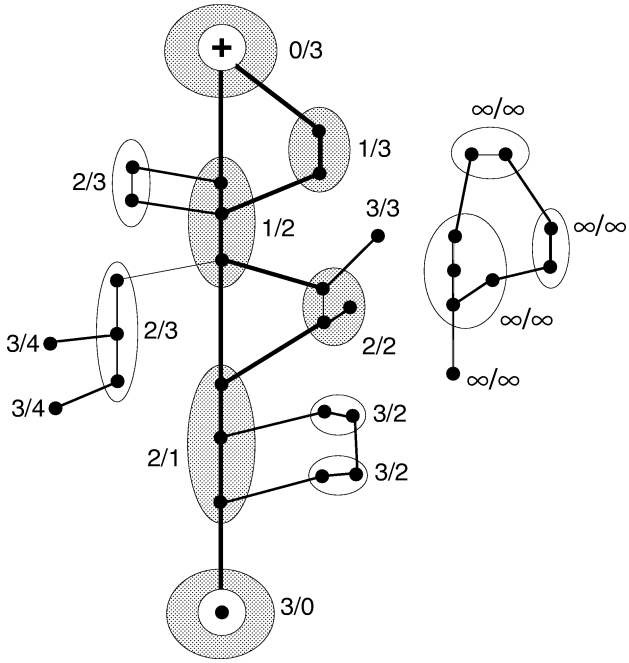

Fig. 4. Supernodes in the circuit of Fig. 3.

supernodes are disjoint, i.e., any node can be a member of at most one supernode, and the f/r values of a supernode are the same as those of all its members. Fig. 3 shows a sample circuit which consists of a set of wires, resistances and switches in a given configuration. After processing, Fig. 4 shows how all edges of zero resistance (i.e., the wires and closed switches) are inside supernodes while the load resistances form the links between supernodes. If we ignore the member nodes and consider the graph formed from the supernodes we obtain a clearer view of the circuit configuration (the active parts are emphasised). We also notice in Fig. 4 how the isolated circuit section is identified by its ∞ labels.

5.2. Node labels define node voltages

From the f/r node values produced by **label-path-resistances** a number of very useful pieces of circuit information can be gleaned without any further analysis. In particular, all nodes can be labelled with their qualitative voltages. We remember that current can only be assigned internally to components, i.e., edges in the graph, while voltage can only have meaning at the terminals, i.e., the nodes. As there are three qualitative resistance values, there are 9 possible labellings of a circuit node. These are shown in Table 7 together with their corresponding qualitative voltage.

First, any node which has ∞ for both forward and reverse labels is not connected to either of the supply terminals and hence must be electrically dead. Thus these nodes can be

Table 7
Meaning of f/r resistance values at a node

Node value		Interpretation		
Forward	Reverse	Condition	V	I
0	0	Short across supply	$\sim$	$+$ for some adjacent edges
0	$+$	Active path through node	$+$	$+$ for some adjacent edges
0	∞	Dead node	$+$	0 for all adjacent edges
$+$	0	Active path through node	0	$+$ for some adjacent edges
$+$	$+$	Active path through node	$\sim$	$+$ for some adjacent edges
$+$	∞	Dead node	$+$	0 for all adjacent edges
∞	0	Dead node	0	0 for all adjacent edges
∞	$+$	Dead node	0	0 for all adjacent edges
∞	∞	Dead sub-graph	$\emptyset$	0 for all adjacent edges

assigned $\emptyset$ for voltage and we note that all adjacent edges will have 0 current flow. Nodes which have an ∞ value for just one of the f/r labels also have no current flowing but have a connection to one of the terminals. If the ∞ occurs as the second value then the voltage of that branch should be assigned positive; if the ∞ occurs as the first value then the node is at ground potential. This gives a general rule: if a ∞ symbol occurs at the power terminals then we know that the whole circuit is dead and will not need further processing.

Of the remaining 4 cases, 2 have direct connections to supply potentials, the voltage is therefore known accordingly, and 2 cases are in the middle of an active path. It is important to recognise that both these latter cases could be labelled $+$ voltage and the qualitative scheme would remain 2-valued and would still be consistent with electrical circuit theory. We only introduce new labels to retain extra information associated with particular configurations. For the case of a node labelled $+/+$ there are resistances in both directions from the node and so the voltage is at some value between the supply potentials and is assigned $\sim$. The case of a direct short across the power supply, $0/0$, is a special case which should be immediately identified. We assume that no further processing of power shorts is necessary and will not consider these further.

Having labelled the node voltages, we can make some observations about the possible currents flowing from the nodes. These are incomplete observations because the current values in Table 7 necessarily refer to gross activity at the node and, as Kirchoff's current law for junctions dictates, further analysis is needed to determine individual edge currents.

For any given edge there are 9 possible labellings for each end and 3 values for edge resistance; this gives 243 cases in all. However, as we have seen, the dead nodes determine all their adjacent edge currents and so only three of the end labellings remain unresolved: $+/+$, $+/0$ and $0/+$. This gives 9 possible cases for each edge resistance R. Taking these in turn, if $R = \infty$ there is no flow in the edge for all 9 cases. If $R = +$ then there will be flow for 7 cases and no flow for the 2 cases where both ends have a 0 in the same path resistance variable. Finally, if $R = 0$, then there are only three cases, as the labels must be the same at each end of the edge, and I can take any value for all three cases.

It is important to notice that all edges between nodes marked $+/+$, i.e., at intermediate voltage, are assumed to carry current flow. In fact, this will be incorrect if the edge is part of a balanced bridge circuit as the flow could then be zero. It is impossible for *any* qualitative representation to resolve the balance of a bridge circuit as the magnitude and direction of flow depends entirely on the exact numerical ratio of the associated resistance values. Even finer resolutions of the quantity space, such as order-of-magnitude representations, would be of little use because numerical equality is necessary to determine precise balance conditions (but see [17] for experiments with order-of-magnitude resistance values that are able to resolve more cases). We assign positive flow in such cases because, as Section 7 discusses, this satisfies the FMEA requirements in all but the most contrived situations.

To summarise: all node voltages are known; of the 9 node labellings 5 cases signify dead edges with zero current; one case is a supply short-circuit; of the remaining 3 cases there are 21 possible end label combinations for edges in which we know their currents in 18 cases without ambiguity. This shows that the constraints of electrical physics helpfully reduces the labelling options quite severely. However, there still remain important topological conditions for inactivity which must be identified by further processing.

5.3. Phase 2: The block finder

Returning to the task requirements we see that any active circuit will consist of a series of active paths between the supply terminals and possibly some additional connected sub-graphs that are inactive due to being either shorted-out or having dead-ends. Removing these cases will identify the main active circuit and it is the purpose of the block finder to perform this graph-theoretic function. Fig. 4 shows some examples of paths ending in dead-end nodes (of degree 1), sub-graphs joined by single edges (sometimes known as an *isthmus*), and sub-graphs joined by several edges to a single active (super) node. In Fig. 4 the active parts are the 6 shaded supernodes and 7 associated load resistors, leaving 13 resistors, 4 nodes and 7 supernodes inactive.

It is an implicit graph-theoretic property of active electrical circuits that every node must have two disjoint paths to the power terminals. If we join the power terminals, by connecting them to a special additional supernode, then the active part of the circuit is a 2-connected graph in that it cannot be disconnected by removal of less than two nodes. (Removing one node may stop current flow but at least two are needed to break the circuit into two parts.) Such 2-connected graphs are called *blocks*. Any dead-end structures will be joined to the active circuit block by a single *cut node* or *articulation point*. Remembering that any shorted-out branch will have both ends contained within *the same* supernode, we see that these sub-graphs also will be joined to the active parts by a single (super) node. We can thus deal with both types of inactivity by using a graph-theoretic algorithm which locates the blocks in a circuit. All edges in the block containing the power supernode are labelled active and all other blocks are inactive. The block finding algorithm is as follows:

procedure *find-active-blocks* (circuit-graph)
let node-count := 1
let $U :=$ the set of all supernodes (S) and the remaining nodes ($T - S$)
forall nodes $v \in U$ let index(v) := 0

```
create a power supply supernode, p, and add extra edges to ⊕ and ⊙
block-finder(p)
end-find-active-blocks

procedure block-finder (s)
let index(s) := node-count
let low(s) := index(s)
let node-count := node-count + 1
for each node or supernode w ∈ adjacent(s) do
    push edge (s, w) onto stack
    if index(w) = 0 then do
            back-marker(w) := s
            block-finder(w)
            if low(w) ⩾ index(s) then
                                empty stack up to (s, w)
                                label the unstacked edges 0
            low(s) := min(low(s), low(w))
    else if w ≠ back-marker(s) then low(s) := min(low(s), index(w))
end-block-finder
```

The algorithm runs on the supernodes, rather than their members, and starts at the power supply supernode in order to label that block last. Thus all blocks found except the last are inactive and are labelled accordingly.

The final CIRQ solution process for **Solve** is thus as follows:

(1) Label the nodes with their forward path resistance: **label-path-resistances**($\oplus$, E, forward)
(2) Label the nodes with reverse path resistance: **label-path-resistances**($\odot$, E, reverse)
(3) Label all nodes with voltage labels (using Table 7)
(4) Label any dead branches (edges from nodes with ∞ labels) with current $= 0$
(5) Find all isolated blocks and label with current $= 0$: **find-active-blocks**(E)
(6) The final block is the active block so label all its edges with current $= +$

Many circuits have separable major branches, i.e., distinct sub-graphs that are only joined via their connection to the power terminals. If these exist, a useful device for enhancing the efficiency of the analysis process is to apply **Solve** separately to each major branch thus partitioning the problem and gaining early information on the status of each individual section.

6. A small illustrative case study

In order to access the methods described we now examine their role in the FMEA task. A small section of a circuit is shown in Fig. 5. This consists of a relay switched pump with associated fuses and an indicator lamp. Realistic applications contain hundreds of components and we do not have space for their voluminous FMEA output but a small example will illustrate all the main features of our qualitative method for failure analysis. Other examples can be found in [15] and [16].

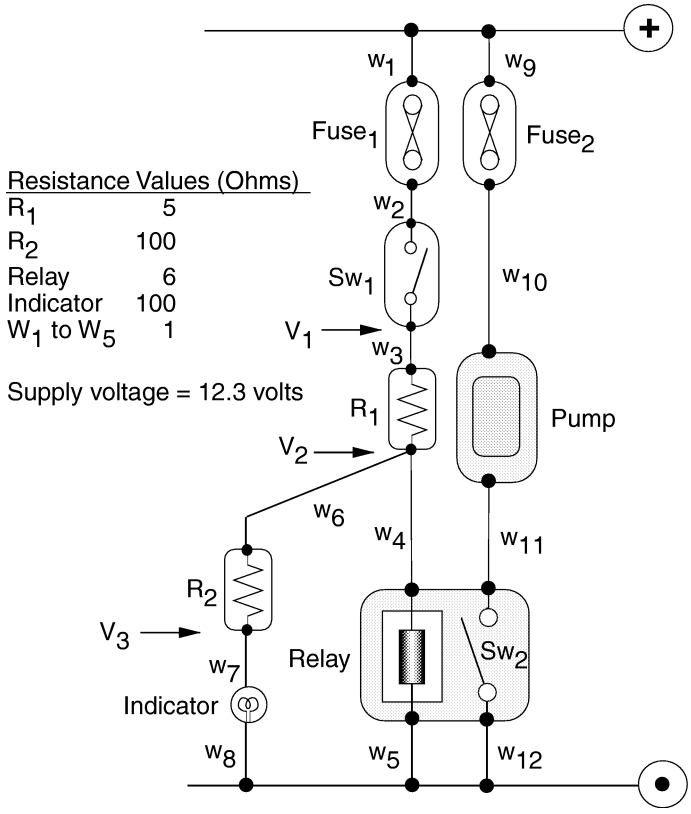

Fig. 5. A simple pump circuit.

The circuit consists of two separable major branches from the power supply. We can treat these independently and for illustration purposes we gain little by including the pump branch. Thus, in much of what follows we will ignore the pump branch.

6.1. The FMEA schedule

We apply the fault classes described in Section 2.3. For open-circuit faults notice that there is a choice for the engineer; candidates for faults could be just the connecting wires or they could include all components. We consider all components as candidates and these are shown in Fig. 6 where the edges marked ℓ are the load edges and all others are of resistance zero. The structure of this circuit consists of three serial chains. This raises another choice for the FMEA schedule. In terms of current flow, an open-circuit fault at any point on a chain is indistinguishable from any other point; however in terms of node voltage there will be differences. We are only interested in current flow and for this there are only three different fault application points possible and so we select three candidate points, one on each distinct branch: say W_3, W_4 and W_7. For short-to-supply faults we have two cases and, again, we can eliminate any redundant tests. Thus, candidate points for short-circuits

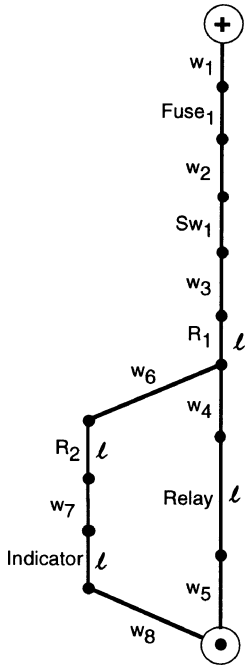

Fig. 6. The graph E for the pump circuit.

to $\oplus$ are the nodes at one end of the wires: W_2, W_3, W_4 and W_7. Candidates for shorts to $\odot$ are exactly the same points. The three nodes V_1, V_2 and V_3 identified in Fig. 5 are test points for measuring voltages.

6.2. Numerical analysis

First we apply a conventional numerical simulator to the system. There are two normal mode states; Sw_1 open or Sw_1 closed. As the whole circuit is dead for the former we only need to compute the latter case. For an example fault we chose the short to positive supply at wire 7 (test point V_3). The currents in the main components and the voltages at the measurement points for the sample fault and the normal mode are shown in Table 8.

The currents in R_1 and the relay have changed by very small amounts (about 4%) while the indicator and R_2 show much larger changes, although these current magnitudes are still much lower than for the relay branch. The voltages do not offer much information; the only

Table 8
Numeric FMEA output

Mode	Current—(amps)				Voltage—(volts)			
	R_1	Relay	R_2	Ind	V_1	V_2	V_3	$V_\oplus$
Normal, $Sw_1 =$ on	1.040	1.010	0.030	0.030	11.260	6.059	3.029	12.3
Fault, short $\oplus$ to V_3	0.995	1.055	0.060	0.123	11.295	6.329	12.3	12.3

large change is at the fault point. From this we see that the differences between normal and fault cases are often unsatisfactory as indicators of effects, indeed it is quite possible for all differences to be almost identical thus offering no information whatsoever. In order to identify significant changes **Diff** must calculate *percentage changes* or some other form of ratio measure. This is also important for defining when a trickle current should be treated as zero current (although this case is not shown in this example). As mentioned before, the level at which these ratios become significant will vary with location in the circuit and so, in order to automate **Diff**, a large number of *different* thresholds will need to be identified and specified. In contrast, the qualitative version of **Diff** singles out R_2, the indicator and V_3 as the items mainly effected by this fault.

6.3. Qualitative analysis

The results for the case study using CIRQ for all faults are shown in Table 9. See Fig. 7 for the node labels and supernode structure. The choice of nodes for tests is made easy by the supernodes as described. In these trials we exclude the power supernode and its two linked supernodes from short circuit faults, as this would produce power short conditions that have little interest. Consequently, the supernodes of interest are only SN_2 and SN_4.

Rather than just returning $+$ for an active edge we record the larger of the node resistances, *NR*, at its two terminals. This feature allows **Diff** to give more information, additional to the activation changes, as any path resistance change will also be reported. **Diff** is implemented by computing the differences from the normal mode case: an increase in node resistance is shown as $-$ (to signify likely decrease in current), a decrease is shown as $+$, a deactivation is shown as D and an activation as A. Table 10 shows this and the effected components are clearly highlighted. For the voltages, changes are indicated $+$ or $-$ if they increase or decrease. Comparing with the numeric results for short to W_7 (i.e., $\oplus$ to SN_4) we see only two currents are reported as changed, with the indicator having the most noticeable change. Fig. 8 shows the altered circuit due to this fault and the reduction in the number of supernodes as SN_4 becomes absorbed into SN_1.

Tables 9 and 10 show two options for voltage measurement at open circuit faults because such breaks effectively produce two nodes where previously there was only one.

Table 9
Output from **Solve**

Mode	R_1	Relay	R_2	Ind	V_1	V_2	V_3
Normal, $Sw_1 = \text{off}$	0	0	0	0	0	0	0
Normal, $Sw_1 = \text{on}$	2	2	3	3	$+$	$\sim$	$\sim$
Short, $\oplus$ to SN_2	0	1	2	2	$+$	$+$	$\sim$
Short, $\oplus$ to SN_4	2	2	2	1	$+$	$\sim$	$+$
Short, $\odot$ to SN_2	1	0	0	0	$+$	0	0
Short, $\odot$ to SN_4	2	2	2	0	$+$	$\sim$	0
Open, W_4	3	0	3	3	$+$	(0 or $\sim$)	$\sim$
Open, W_7	2	2	0	0	$+$	$\sim$	($\sim$ or 0)

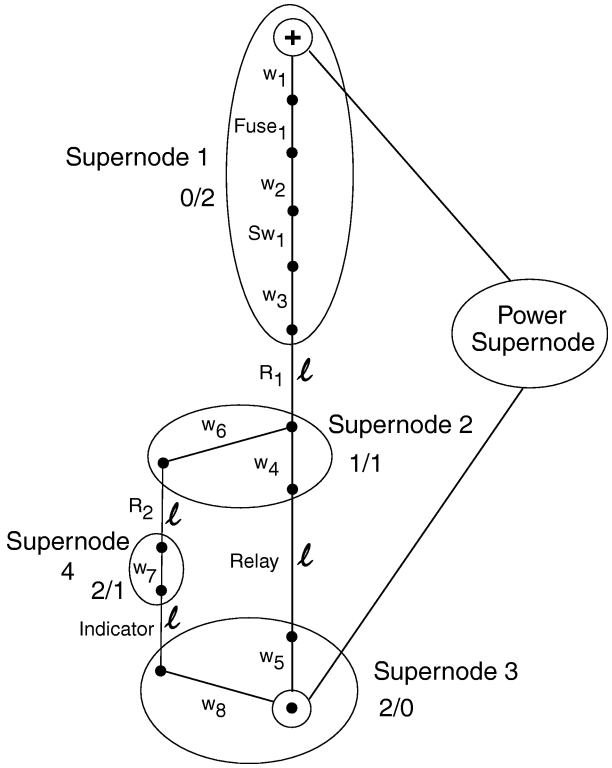

Fig. 7. Supernodes in the pump circuit.

Table 10
Output from **Diff**

Mode	R_1	Relay	R_2	Ind	V_1	V_2	V_3
Normal, $Sw_1 = \text{on}$	2	2	3	3	+	∼	∼
Short, ⊕ to SN_2	D	+1	+1	+1		+	
Short, ⊕ to SN_4			+1	+2			+
Short, ⊙ to SN_2	+1	D	D	D		−	−
Short, ⊙ to SN_4			+1	D			−
Open, W_4	−1	D				(none or −)	
Open, W_7			D	D			(none or −)

It is important to remember that the changes being indicated are not electrical values but changes in path resistance. In the case of the short to W_7 there are no activation changes, i.e., the qualitative results are the same for both fault mode and normal mode (all components active). However, the topological alterations due to the new current injection causes changes to the local resistance paths which are reported.

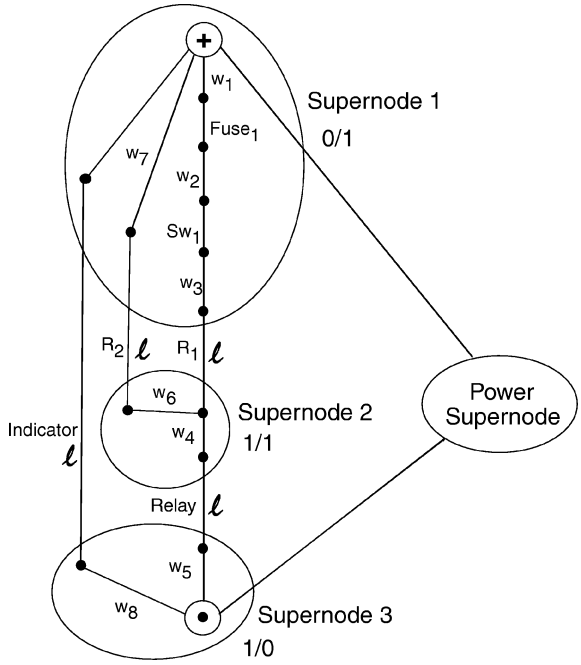

Fig. 8. The result of a short-circuit to Wire-7.

7. Satisfying the FMEA task requirements

First we observe that the method uses algorithms that are efficient and correct. The complexity of **label-path-resistances** and **find-active-blocks**, for the graph $E(T, R)$, is $O(|T|^2)$ and $O(|R|)$, respectively, and proofs of their correctness in assigning labels and removing blocks are available in standard texts, e.g., [5].

The specific requirements for our FMEA task state that a component change list is to be produced to record which components have changed their electrical state. It is sufficient to indicate only changes from presence to absence of flow or vice versa and the direction of flow is not essential. We now consider the soundness and completeness of the method.

Considering the FMEA defined fault classes, open-circuit faults will change the local current status of any active path and will also alter node voltages, where appropriate, in inactive paths. These effects in the qualitative model clearly mirror identical behaviour in the quantitative world. For short-circuit faults we consider the junction currents at affected nodes; for any given active node there will be a set of input currents and a set of output currents, both sets being non-empty. When a short is applied (say to positive) then the node voltage is raised to the positive supply level and all current then flows in from the new connection. Thus any edges previously supplying the input set must now either reverse their flow direction and take output current or will be shorted out and therefore take zero current. In the case of CIRQ the affected node will also be raised to the positive terminal, as its f value changes to 0, and any incident branch flowing from a positive source will then

become inactive. Hence, any power short-circuit conditions will be represented by changes of node and path resistance values and any shorted-out components will be detected by **find-active-blocks**. In the case of reversed flow direction it is possible that no shorts occur and therefore no changes of activation are detected. This is in accord with the real circuit and FMEA does not strictly require reports on these conditions. Nevertheless, the affected node will always suffer some path resistance disturbance (unless it is already at the supply voltage, in which case the fault would have no effect anyway) and these will be recorded as path resistance changes by **Diff**. We see that our models capture all changes in topology and major changes of resistance localised with the fault but not minor or indirect resistive changes. The method is complete in that any open-circuit or short-to-power fault will always be detected in the form of node resistance changes and the edges directly affected will be identified. It is incomplete in that incremental changes in resistance due to indirect or remote topological changes cannot be captured in the model.

We notice that this is only true for shorts to power terminals; the more general case of shorts between any two arbitrary nodes could produce quantitative resistance changes that are not captured in our qualitative representation. It is a reasonable hypothesis that such changes are also difficult to detect by human analysts and this may be another reason why FMEA is not usually performed on such fault classes.

The case of bridging elements requires examination. It is clear that SP reducible circuits will be correctly analysed by **Solve** and directional information is also available, see Section 9. This is also true for many of the edges in non-SP reducible circuits except for the bridging edges in which both flow direction and magnitude are ambiguous. The circuits in Fig. 1 are examples where the flow (and direction) can be labelled in the unmarked (non-bridge) edges but are ambiguous in the marked (bridge) edges. It is possible to use further algorithms to locate and label all bridges as ambiguous but this still cannot resolve the balanced case. This is a problem for *all* qualitative resistance representations. It is impossible to deduce the inactive state of a bridge element without quantitative data. (In other work we have experimented with an order-of-magnitude value set for resistance and shown how this can alleviate the problem by resolving many cases of unbalanced (active) bridges [17].)

Consequently, our method is sound in that unrealistic cases are not generated and it is complete for SP reducible circuits. Although being incomplete for non-SP reducible circuits, any change in topology that causes an activity change in non-bridges or a new local power supply connection will be reported in the output from **Diff**. Indeed, **Diff** indicates the location of all major changes and will never give an empty change list when one of the specified fault classes has an effect on an active circuit. A corollary of this is that the status of a bridge cannot change without **Diff** producing a non-empty change list.

The effects of the incompleteness for bridges are minimised for FMEA by the approach adopted in CIRQ. All bridges are labelled as active by **Solve** which assumes the balanced case does not exist in the circuit being processed. In practice zero flow cases are very rare as exact balance requires the fine tuning of at least four associated circuit elements. Hence most bridges in real circuits are active (i.e., unbalanced) and balanced cases are designed for special functions, (they will be identified for special treatment in the design documents). In practical applications of CIRQ in automotive FMEA, Snooke and Price report that 85% of the circuits encountered could be handled immediately with the remaining 15%

requiring special treatment to deal with complex behaviour [31]. There were no cases of balanced bridges and no other directional flow problems that could not be resolved. Consequently we use the heuristic that it is extremely unlikely that a fault would cause the resistances to change to precisely the right value for a bridge to become balanced (unless balance was included in the original function of the circuit and this would be known and isolated). We have found very few cases of balanced circuits in real applications and find the above safeguards sufficient for much practical FMEA work.

8. Further aggregation algebras

The node path resistance variables hold local values of some global or semi-global properties of the circuit. We have examined one formulation but we can ask if there are any other algebras for combining resistances that provide interesting results. From circuit theory we remember that all configurations have a dual and conductance is the dual of resistance. Conductance is the reciprocal of resistance and therefore the relationship between qualitative resistance R and qualitative conductance G is as shown in Table 11.

This means that Min R is exactly equivalent to Max G and we can replace Min R with Max G in Table 3 and obtain exactly the same qualitative results. The assignment table for serial and parallel circuit reduction using this formulation is shown in Table 12.

Different integer labels may be produced but the relation between R and G does not conflict with the requirements of Section 4.2. By this means we can employ conductance values to obtain alternative algebras for combining resistances. (For consistency, we show

Table 11
Qualitative resistance and conductance

R	0	ℓ	∞
G	∞	ℓ	0

Table 12
Aggregation using Sum R and Max G

A	B	A & B in series	A & B in parallel
∞	∞	∞	∞
∞	m	∞	m
∞	0	∞	0
n	∞	∞	n
n	m	$n+m$	$\max(n,m)$
n	0	n	0
0	∞	∞	0
0	m	m	0
0	0	0	0

Table 13
Summary of aggregation algebras

A	B	Series(A, B)	Parallel(A, B)	Effect
n	m	Sum R	Min R	Minimum path length
n	m	Sum R	Max G	Maximum path length
n	m	Min G	Sum G	Minimum cut-set
n	m	Max R	Sum G	Maximum cut-set

all values as resistances in this section although the results have been derived by using conductance for some rules.) The reduction rules in Table 12 are identical to previous except for line 5 and as this line still returns a positive integer the qualitative interpretation is unchanged. However, for parallel resistive edges the Max G function returns larger values and these propagate through the network so that the algebra identifies the "paths with the maximum number of load resistances". By traversing from a supply terminal and following nodes with the same node resistance value, i.e., $NR(\oplus)$, it is possible to locate the nodes on a "longest" path.

Another interesting case occurs when using Min G for series and Sum G for parallel. These rules assign values to nodes that represent "the number of paths leaving the node on route to a supply terminal". If a node has f/r values of n/m then there are n different paths from this node to the positive supply terminal and m paths to the other terminal. With this scheme the supply terminals will often have different values. From this node data the minimum cut-set needed to deactivate the circuit can be found.

Finally, [10] the dual of Min G gives Max R for series and we combine this with Sum G for parallel. The node results now give the "number of distinct (exclusive) paths from a node to the supply". Due to the many combinations of parallel circuits this scheme produces quite large numbers, especially at the power terminals. See [15] for illustrated examples of these labelling schemes.

These results are properties of the node resistance idea in which the two local variables reflect some relevant global topological features of the circuit graph. Table 13 summarises the four cases. The first two are "depth properties" in that they measure maximum and minimum path lengths between the supply points. The last two are "breadth properties" in that they record maximum and minimum "widths" of the circuit.

The different circuit criteria have different benefits for specific applications. We have found the Sum R, Min R algebra most useful, mainly because of its ability to emphasise direct paths in a way that is suitable for FMEA. However, the path with the maximum number of active load nodes may be of interest when diagnosing a faulty active circuit and both the minimum number and the maximum number of paths between selected nodes may be important in cases of network analysis, e.g., for redundant routing when sections are disrupted.

[10] Other combinations such as Sum R and Sum G do not produce useful labellings as using the same function does not distinguish between the different topologies.

9. The benefits and limitations of qualitative resistance models

The motivation for this work has been (a) to produce efficient algorithms for repetitive FMEA processing and (b) to explore qualitative models in terms of minimal representations. The former has been satisfied by the algorithms **label-path-resistances** and **find-active-blocks** which are efficient and correct. In terms of electrical circuit theory, our method is sound in that unrealistic cases are not generated and it is complete for voltage labelling. It is complete for current labelling for SP reducible circuits but incomplete for balanced bridge conditions. All local activity changes due to the specified fault classes in non-bridging branches are always reported and node resistance changes give valuable additional indications of flow disturbances.

The supernode concept provides considerable advantages for FMEA fault analysis. Two particularly appropriate features concern the identification of sites for fault application. First, we no longer need to apply the "short-circuit to power" fault to every node in the circuit but only to each supernode. This is because all nodes within a supernode will give exactly the same result—supernodes are groupings of electrically equivalent nodes. Secondly, many circuits contain sequences of wire runs that simply connect two nodes through a series of intermediate nodes (all of degree 2)—such conducting series circuits become allocated to a supernode and can then be treated as a single entity. This is useful for FMEA open-circuit faults, as any supernode of degree 2 containing n nodes only requires one fault application rather than $n - 1$ applications. These observations serve to reduce the size of the fault space and the associated fault change list.

Regarding the minimal nature of our representation, resistive networks may seem very restrictive as modelling tools but, in fact, they have wide applicability. Other physical systems have equivalent flow and potential variables acting on similar networks of relations. These include mechanical, hydraulic and other systems in steady-state, all of which can be modelled in the same resistance/voltage/current framework. This strong analogy between different types of physical and engineering systems has been exploited in many fields. Indeed, we have used CIRQ to model the mechanical drives, gears and mechanisms that were actuated by the electric motors in a circuit [23]. Here torque takes the role of voltage and rotational motion replaces current. The output is a list of which mechanical components are moving or stationary. The application of such an ontology to different situations is the principle behind unified modelling approaches such as System Dynamics and Bond graphs [14]. The advent of highly integrated mechatronic systems has stimulated the need for such general and unified models, with many existing electrical simulators and CAD tools being adapted and extended with different domain libraries for use across various engineering domains [29].

Some of the apparent limitations of our model can be overcome by various forms of extension or alteration. An obvious example is that other forms of steady-state, such as occur in alternating current circuits, could be analysed. We remember that the Laplacian operator can be employed to convert the equations of an alternating current network into the same form and structure for analysis. This has been done and qualitative phasor analysis has been developed [9] by adding an additional circuit variable to record phase angle as well as magnitude.

Other limitations in this minimal qualitative model appear to be more serious: it is time invariant, it has a single power source, and only deals with one type of idealised component. We will explore these potential problems and their solution in more detail:

Non-linear elements. CIRQ is simple because it relies on steady-state analysis and under quiescent conditions non-linear devices can be replaced by equivalent linear models. [11] This simplicity gives the method its power and we must avoid the temptation to enhance the base models with extra complexity in order to cover many more features. The idea of multiple step analysis as described in Section 2.3 provides an attractive way of dealing with time varying effects. Each configuration of a circuit is analysed as a snapshot and then changes are made corresponding to the transition to the next state. This is useful in many situations, for example, we could model capacitors and inductors in this way. Electrical capacitance and inductance are non-dissipating energy storage devices with equations $i = (\mathrm{d}v/\mathrm{d}t)C$ and $v = (\mathrm{d}i/\mathrm{d}t)L$, respectively. Thus, they only have effect when current or voltage is varying and for steady-state DC circuits these devices can be replaced by open circuits, $R = \infty$, (capacitor) or short circuits, $R = 0$, (inductor). For high frequencies they may reverse these roles. If we wish to model at a slightly more detailed level of approximation we can recognise that during step changes both capacitors and inductors act as loads and so $R = \ell$ during $t = 0$ to $t = t_c$ (analysis step 1), and for $t > t_c$ (analysis step 2) the above values are used, respectively. We will need to normalise all capacitors and inductors if we wish to use a single time constant t_c ($= RC$ or L/R) throughout. Hence a two step analysis can be used to treat these components as "time dependent resistors". Various other devices can be treated in the same way and their multi-step specification can be entered into a suitable state table. See [30] for details of this multi-stage analysis using state tables in FMEA. Regarding other dynamic effects we notice that, unlike diagnosis, FMEA is not troubled by glitches, transients or intermittent faults. During design analysis any such fault can be made hard and then simulated as a series of snapshots as required.

Directional elements. For many circuits and especially SP reducible circuits it is possible to produce labels for edge current flow direction using the node resistance values. In previous schemes we have explored path following algorithms that determine branch flow directions [16].

We observe that in an active circuit the labelling at the positive supply node, *0/m*, will always be reflected at the ground node as *m/0* and there will be at least one path in the circuit between the supply terminals for which $NR(n_i) = NR(\oplus)$ for *all* nodes n_i on the path. We denote such paths the *primary* paths.

In SP reducible circuits the primary paths have no short circuited or open circuited edges and all their nodes and edges are always active. All other paths, termed *secondary* paths, contain all the short and open circuits, bridges, ambiguities and other less direct routes.

[11] We have assumed all resistors are linear but, in fact, non-ohmic resistors would make no difference to our qualitative results.

Flow direction can be determined as follows. Let a simple series path exist between end nodes u and v and define

$$F(u) = \frac{f(u)}{f(u) + r(u)},$$

then

$$\text{the direction of flow is:} = \begin{cases} \text{from } u \text{ to } v & \text{if } F(u) < F(v), \\ \text{from } v \text{ to } u & \text{if } F(u) > F(v), \\ \text{ambiguous} & \text{if } F(u) = F(v). \end{cases}$$

This heuristic is useful for building path following algorithms that distinguish primary and secondary paths [16].

A common directional device is the diode which acts as a conductor or an insulator depending upon the direction of applied voltage. The FMEA package built upon CIRQ [30] is able to handle diodes by using a two step analysis. First, the current directions are deduced as above and if in agreement with the orientation of the diode then the result is complete otherwise a second analysis is performed, this time assuming the diode operates in the other direction.

Multiple power sources. A single two-terminal power source is used throughout our work as this satisfies the requirements of the application domain. We have not yet extended this work to multiple source circuits but notice that a difficulty immediately arises for qualitative methods if the sources have different voltage levels. The location of zero current branches would not then depend only on circuit topology and the location of the sources but also on their voltage ratios. We believe the superposition theorem offers the best way of analysing any multiple source circuit. An n source circuit would be applied to CIRQ with each source active in turn and all other sources replaced by zero resistance links. The n results for component activity could then be analysed for consensus or conflict by a multi-way version of **Diff**.

10. Related work

The application and development of techniques for reasoning about electrical and electronic circuits is not new in AI. The complex tasks of design and diagnosis in particular have provided motivation for developing methods and tools that can help the engineer to manage the many facets of this knowledge-rich domain.

Of the early work, outstanding contributions have been made by Johan de Kleer following seminal papers by Gerald Sussman and colleagues. In [36] Sussman introduces the issues characterising circuit design as an AI problem, and in [32] Stallman and Sussman describe a rule-based constraint propagation system. This approach was used by de Kleer and Sussman to implement an interactive design tool that incrementally tackles parts of circuit design problems [7]. Circuit laws were represented in *constraint expressions* that, together with local variables, form a network that models the circuit under study. The method is symbolic and this enables the relations between the results and the design

requirements and constraints to be kept transparent. Thus, de Kleer and Sussman argued, the solutions are insightful, unlike numerical methods that are difficult to interpret and reconcile with design desiderata.

In a major paper on commonsense understanding of circuits de Kleer [6] gave a theory of intuitive reasoning based on qualitative models. A computer program based on this theory was able to produce qualitative causal analyses. The circuit models used piecewise-linear approximations and performed first-order incremental analysis. This required a treatment of time which took the form of state transitions. The complexity and subtlety of electrical circuit concepts is illustrated by de Kleer's analysis of teleology which showed that a single resistor can have no less than 18 possible functional roles in a circuit.

Other temporal qualitative analysis systems include the work of Williams [40] who used sophisticated models of transistors for digital and analog electronics. More recently there have been ECAD tools developed that employ qualitative reasoning, for example, the work of [19] which uses qualitative variables to model approximations that cannot be managed by numeric equation solving systems alone. In circuit design with ECAD tools there is an iterative feedback process: produce tentative design; simulate to obtain performance data; correct the design by modification; simulate again; etc. Numeric data are too precise to capture useful rules for use in trading-off design requirements and generating effective corrections. By using qualitative values for reasoning about the corrections to be applied [20] the number of simulations needed is markedly reduced (often to only one).

The decomposition of circuits into series/parallel sub-graphs has been used by Flores and Farley [9] to analyse alternating current linear circuits in steady-state. They also used constraint-based models and *phasors* to handle sinusoidal variables. Phasors are representations, very like vectors, that capture the phase angle as well as the magnitude of a quantity. By shifting to the complex domain allows solutions to be found in terms of simultaneous algebraic equations and then the same methods apply as for non-alternating steady-state systems. This has been applied to incremental design [10] in order to find suitable modifications to a circuit to meet additional design goals.

A research group in Munich have explored our original f/r labelling concept (first described in [15]) and built similar algorithms for applications in diagnosis and FMEA [34]. They report on a circuit model variation in which each component terminal has four connectivity variables: two for the resistance value to each supply terminal on *external* paths from the component (out paths) and two for resistance values to each supply terminal on *internal* paths that pass through the component (in paths) [35]. This appears to give more information about components in terms of extra constraints. Comparison shows, however, that the methods give identical results and the representations are mathematically equivalent. Given that all components consist of single edges of resistance R with a node at each end, say i and j, then let $f_{out}(i)$ be the external path resistance from i to $\oplus$ and let $f_{in}(j)$ be the internal path resistance from j to $\oplus$. Then these are related by the edge resistance:

$$f_{in}(j) = f_{out}(i) + R \quad \text{and} \quad f_{in}(i) = f_{out}(j) + R$$

(and similarly for the r labels). As R is known we can always derive the internal paths from the external values and so half of the connectivity variables are redundant. Furthermore, as the qualitative values used are $[0, +, \infty]$, differences in connectivity to power terminals can only be distinguished as either 0 or $+$. The problems with such limited connectivity were

recognised in [35]—this whole topic often raises confusions due to the subtle relationships between local and global properties of electrical network theory.

A more promising variation would be to maintain an f/r label for *each* edge incident at a node. These would hold the path resistance values for the end of an edge to each power terminal, respectively, *as if the edge was disconnected from its node*. This would require a more complex labelling algorithm (and would need $4|R|$ labels rather than the present number of $2|T|$) but would allow further useful inferences about path activity to be gained, including the identification of dead-end branches. It can be seen that the f/r node values in our present system are the minimum values of all such incident edge labels at a node.

Other work on qualitative circuit analysis has similarly shown the problems of propagating constraints with local variables. In order to deal with bridge circuits, Mauss and Neumann have argued that the standard series/parallel reduction rules are insufficient and that the star/delta transforms must be used to resolve such cases [21]. By repeated applications of series/parallel/star/delta (SPS) replacement rules [12] any resistive mesh can be converted into a single equivalent resistor. The total current can then be calculated and, through reverse decomposition, the currents in each edge can be assigned. During the reduction down to a single resistance Mauss and Neumann record the decomposition process in a SPS reduction tree. This tree is then used to derive the values for the circuit edges during the reverse process. A constraint network is also produced for controlling the local propagation of constraints. The constraint network is very similar to that used in [7] to manage constraint expressions but this is not surprising considering the commonality of purpose evident in much work on circuit analysis. This method has the advantage that either symbolic or numeric values (or both) can be used in the process. An interesting (numerical) extension employs a general resistance model as a line in the VI plane which can then incorporate additional voltage sources [22].

For qualitative labelling, SPS reduction gives no advantages over our method and the method of Mauss and Neumann, although more complicated, delivers qualitative results exactly equivalent to those from our algorithm. As we have shown, it is incorrect to believe that star/delta reduction rules or other transforms are necessary to label SP irreducible circuits. All qualitative formulations may encounter bridges that must be returned as ambiguous cases (as does [21]) and traversal algorithms like **label-path-resistances** are efficient in navigating any topology including bridges and multi-way star/delta circuits.

An interesting application of infinite valued resistors is in [24] where insulators are modelled between all pairs of conductors that might short together if a soldering fault occurs on a circuit board. A diagnostic system can then postulate that an insulator changes its value to represent a fault and thus introduce additional constraint into the system.

11. Summary

The qualitative model of electrical circuits presented here has proved effective and efficient in supporting automated failure reasoning in a particular FMEA environment. It satisfies our specification of the FMEA task as open-circuit and power short-circuit

[12] The theory of partial 2- and 3-trees covers series/parallel and star/delta reduction, see [2].

faults are qualitative changes which have topological interpretations that are correctly reflected in the model. The CIRQ algorithm is sound and complete in that for a single source resistive network with qualitative resistance values $[0, \ell, \infty]$ all nodes and edges are labelled for qualitative resistance and voltage. For SP reducible circuits, the current labelling is complete and it is also possible to assign directions of flow for most edges. For a general (SP irreducible) resistive mesh directional labelling in qualitative circuit models will always be incomplete because certain branches (bridges) will have ambiguous flow direction and may therefore be balanced so that no current flows. Our method labels all such branches as active and we have shown how this will satisfy the FMEA requirements in all but the most contrived cases.

We have seen how qualitative resistance values of circuit edges map into graph-theoretic meaning: the value ℓ can be viewed as a connection between two nodes, the value ∞ is equivalent to no connection and the value 0 indicates that the two nodes are electrically identical. By capturing these local properties the model is able to deduce some global connectivity measures, particularly in relation to the source terminals. Indeed, the CIRQ algorithm is successful because the qualitative formulation equates equivalent resistance reductions to the graph-theoretic property of minimum path length. It is important to remember that paths are compared on the basis of the number of load resistances they contain and not on their actual resistance. We must not view the results as resistance value changes, therefore, but as reflecting changes in circuit topology.

The concepts of supernodes and integer path resistance have proved valuable. Supernodes group together all electrically identical nodes, thus simplifying both fault application and analysis. Integer resistance values allows disturbances to be detected when there are no qualitative activity changes. The integer path algebra is a form of overloading of the resistance value ℓ and other aggregation schemes are possible providing they maintain the basic ordering relation $0 < \ell < \infty$. For example, resistance values with orders-of-magnitude separation allow the lowest valued resistance path to be found rather than the path with the least number of resistive edges [17]. The use of such finer resolution can thus help to deal with the problem of balanced bridges.

The method has a layered approach that provides some indications of gross activity at an early stage. As soon as **label-path-resistances** has terminated the existence of any gross conditions (total power short or dead circuit) is identified and the voltages of all nodes are assigned. Also any circuit branches disconnected from the power supply are labelled. Such information may be of immediate use and therefore remove the need for the following stage for dead-end and shorted-out sub-graph removal. This might apply in related tasks such as design modification when particular parts of a circuit or individual components are the centre of interest.

In this study of qualitative models of electrical systems as resistance networks we found that the need to identify and distinguish different qualitative circuit properties focussed on topological features and inevitably involved graph-theoretic notions. The output from our method provides apposite information for FMEA and forms the basis of real FMEA application systems. To avoid the difficulties of interpreting the complex data space of numeric solutions our model captures the essence of component properties so that the output is more directly focussed on the needs of the task requirements. This more meaningful and "user friendly" output contrasts with that produced by existing

numerical simulators and suggests that abstract models have value in many stages of engineering reasoning not just during the early stages of design or analysis. We have attempted to place this work in context and clarify some of the issues that arise from the subtle relationships between the local and global properties of electrical circuits. We hope the qualitative techniques described here offer a contribution towards the development of further automated tools that will display increasing empathy with the domain concepts and intuitions used by engineers.

References

[1] S. Arnborg, B. Courcelle, A. Proskurowski, D. Seese, An algebraic theory of graph reduction, in: H. Ehrig, H.-J. Kreowski, G. Rozenberg (Eds.), Graph Grammars and Their Application to Computer Science, Lecture Notes in Computer Science, Vol. 532, Springer, Berlin, 1991, pp. 70–83.
[2] S. Arnborg, A. Proskurowski, Canonical representations of partial 2- and 3-trees, in: J.R. Gilbert, R. Karlsson (Eds.), Proc. 2nd Scandinavian Workshop on Algorithm Theory, Lecture Notes in Computer Science, Vol. 447, Springer, Berlin, 1990, pp. 311–319.
[3] P. Brna, A. Caiger, The application of cognitive diagnosis to the quantitative analysis of simple electrical circuits, in: C. Frasson, G. Gauthier, G.I. McCalla (Eds.), Intelligent Tutoring Systems, Proc. 2nd International Conference, Springer, Berlin, 1992, pp. 405–412.
[4] W. Clocksin, A.J. Morgan, Qualitative control, in: Proc. ECAI-86, Vol. 1, 1986, pp. 350–356.
[5] T.H. Cormen, C.E. Leiserson, R.L. Rivest, Introduction to Algorithms, MIT Press, Cambridge, MA, 1990.
[6] J. de Kleer, How circuits work, Artificial Intelligence 24 (1984) 205–280.
[7] J. de Kleer, G.J. Sussman, Propagation of constraints applied to circuit synthesis, Circuit Theory and Applications **8** (1980) 127–144.
[8] E.W. Dijkstra, A note on two problems in connection with graphs, Numer. Math. 1 (1959) 269–271.
[9] J. Flores, A.M. Farley, Qualitative phasor analysis, in: Proc. 10th International Workshop on Qualitative Reasoning (QR-96), Stanford Sierra Camp, CA, 1996, pp. 43–51.
[10] J. Flores, A.M. Farley, Incremental design for linear circuits, in: Proc. 12th International Workshop on Qualitative Reasoning in Physical Systems (QR-98), Cape Cod, 1998, pp. 50–56.
[11] K.D. Forbus, Qualitative physics: past, present and future, in: D. Weld, J. de Kleer (Eds.), Readings in Qualitative Reasoning about Physical Systems, Morgan Kaufmann, San Mateo, CA, 1990.
[12] M.J. French, Invention and Evolution: Design in Nature and Engineering, Cambridge University Press, Cambridge, 1988.
[13] C. Inglis, Modern Engineering, Oxford University Press, Oxford, 1955.
[14] D.C. Karnopp, D.L. Margolis, R.C. Rosenberg, Systems Dynamics: A Unified Approach, 2nd ed., Wiley, New York, 1990.
[15] M.H. Lee, A.R.T. Ormsby, Qualitative modelling of electrical circuits, in: Proc. 6th International Workshop on Qualitative Reasoning in Physical Systems (QR-92), Edinburgh, 1992, pp. 147–153.
[16] M.H. Lee, A.R.T. Ormsby, Qualitatively modelling the effects of electrical circuit faults, Artificial Intelligence in Engineering 8 (4) (1993) 293–300.
[17] M.H. Lee, Qualitative modelling of linear networks in ECAD applications, in: Proc. 13th International Workshop on Qualitative Reasoning (QR'99), Scotland, 1999, to appear.
[18] R. Leitch, Q. Shen, G.M. Coghill, M. Chantler, Choosing the right model, IEE Proceedings D, Control Theory and Applications (1999), to appear.
[19] C.A. Makris, C. Toumazou, ISAID: Qualitative reasoning and trade-off analysis in analog IC design automation, in: A. Pugh (Ed.), Proc. IEEE International Symposium on Circuits and Systems, 1992, pp. 1–4.
[20] C.A. Makris, C. Toumazou, Analog IC design automation .2. automated circuit correction by qualitative reasoning, IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems 14 (2) (1995) 239–254.
[21] J. Mauss, B. Neumann, Qualitative reasoning about electrical circuits using series-parallel-star trees, in: Proc. 10th International Workshop on Qualitative Reasoning (QR-96), Stanford Sierra Camp, CA, 1996, pp. 147–153.

[22] J. Mauss, Local analysis of linear networks by aggregation of characteristic lines, in: Proc. 9th International Workshop on the Principles of Diagnosis (DX-98), Cape Cod, 1998, pp. 78–85.
[23] K. Nakata, M.H. Lee, G. Barry, A.R.T. Ormsby, P.L. Olivier, Modelling electromechanical systems from multiple perspectives, in: G. Rzevski, R.A. Adey (Eds.), Applications of Artificial Intelligence in Engineering X, 1995, pp. 227–234.
[24] C. Preist, B. Welham, Modelling bridge faults for diagnosis in electronic circuits, in: Proc. First International Workshop on Principles of Diagnosis, Stanford, CA, 1990; also in Technical Report HPL-ISC-TM-90-036, Hewlett-Packard, Bristol, 1990.
[25] C.J. Price, Function-directed electrical design analysis, Artificial Intelligence in Engineering 12 (4) (1998) 445–456.
[26] C.J. Price, J.E. Hunt, M.H. Lee, A.R.T. Ormsby, A Model-based approach to the automation of failure mode effects analysis for design, in: Proc. Inst. Mech. Eng., Part D: Journal of Automobile Engineering, Vol. 206, 1992, pp. 285–291.
[27] C.J. Price, D.R. Pugh, M.S. Wilson, N.A. Snooke, The Flame system: Automating electrical failure mode and effects analysis (FMEA), in: Reliability and Maintainability Symposium, IEEE, 1995, pp. 90–95.
[28] Saber User's Guide, Analogy Inc., Beaverton, OR, 1996.
[29] J. Schollers, T. Yli-Pietila, Simulation of mechatronic systems using analog circuit simulation tools, in: Proc. IEEE International Conference on Robotics and Automation, 1995, pp. 2847–2852.
[30] N.A. Snooke, Simulating electrical devices with complex behaviour, AI Commun., to appear.
[31] N.A. Snooke, C.J. Price, Challenges for qualitative electrical reasoning in automotive circuit simulation, in: Proc. 11th International Workshop on Qualitative Reasoning (QR-97), Cortona, 1997, pp. 175–180.
[32] R.M. Stallman, G.J. Sussman, Forward reasoning and dependency-directed backtracking in a system for computer-aided circuit analysis, Artificial Intelligence 9 (1977) 135–196.
[33] P. Struss, What's in SD? Towards a theory of modeling for diagnosis, in: Proc. 2nd International Workshop on Principles of Diagnosis (DX-91), Milan, 1991.
[34] P. Struss, A. Malik, M. Sachenbacher, Case studies in model-based diagnosis and fault analysis of car-subsystems, in: Working papers 1st International Workshop on Model-Based Systems and Qualitative Reasoning (ECAI-96), Budapest, 1996, pp. 17–125.
[35] P. Struss, A. Malik, M. Sachenbacher, Qualitative modeling is the key to automated diagnosis, in: Proc. 13th Triennial World Congress of IFAC, San Francisco, CA, Pergamon, 1996, pp. 43–48.
[36] G.J. Sussman, Electrical design: A problem for artificial intelligence research, in: Proc. IJCAI-77, Cambridge, MA, 1977, pp. 894–900.
[37] G.J. Terry, Engineering System Safety, 2nd ed., Mechanical Engineering Publications, London, 1991.
[38] A. Vladimirescu, A.R. Newton, D.O. Pederson, SPICE, Version 2G: User's Guide, Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA, 1980.
[39] D. Weld, J. de Kleer (Eds.), Readings in Qualitative Reasoning about Physical Systems, Morgan Kaufmann, San Mateo, CA, 1990.
[40] B.C. Williams, Qualitative analysis of MOS circuits, Artificial Intelligence 24 (1984) 281–346.