Propositional Logic Diagrams

R.P.Clark

January 9, 2010

Abstract

Propositial Logic Diagrams have been designed to provide an intuitive method for visualising and manipulating logic equations, to express fault modes in Mechanical and Electronic Systems. Diagrams of this type can also be used to model the logical conditions that control the flow of a computer program. This type of diagram can therefore integrate logical models from mechanical, electronic and software domains. Nearly all modern safety critical systems involve these three disiplines. It is intended to be used for analysis of automated safety critical systen Many types of safety critical systems now legally require fault mode effects analysis[**?**], but few formal systems exist and wide-spread take-up is not yet the norm.[**?**]. Because of its visual nature, it is easy to manipulate and model complicated conditions that can lead to dangerous failures in automated systems.

The Diagrams described here form the mathematical basis for a new visual and formal system for the analysis of safety critical software and hardware systems.

1 Introduction

Propositional Logic Diagrams (PLDs) have been devised to collect and simplfy fault modes in safety critical systems undergoing static analysis[**?**][**?**].

This type of analysis treats failure modes within a system as logical states. PLD provides a visual method for modelling failure mode analysis within these systems, and specifically identifying common failure symptoms in a user friendly way. Contrasting this to looking at many propositional logic equations directly in a text editor or spreadsheet, a visual method is percieved as being more intuitive.

PLDs use three visual features that can be combined to represent logic equations. Closed contours, test cases, and lines that link test cases. All features may be labelled, and the labels must be unique within a diagram, however contours may be repeated in the diagram.

Test cases are marked by asterisks. These are used as a visual 'anchor' to mark a logical condition, the logical condition being defined by the contours that enclose the region on which the test case has been placed. The contours that enclose represent conjuction. Test cases may be connected by joining lines. These lines represent disjunction (Boolean 'OR') of test cases.

With these three visual syntax elements, we have the basic building blocks for all logic equations possible.

Test cases - Points on the plane indicating a logical condition.

Conjunction - Overlapping contours

Disjunction - Joining of named test cases.

2 Formal Description of PLD

Definitions of conrete and abstract PLD's follow. Well-formedness conditions for PLD's are separated from this definition, because of practical differences between the way they are used to represent software as opposed to representing electronics and mechanical systems.

2.1 Concrete PLD Definition

A concrete *Propositional logic diagram* is a set of labeled *contours* (closed curves) in the plane. The minimal regions formed by the closed curves can by occupied by 'test points'. The 'test points' may be joined by joining lines. A group of 'test points' connected by joining lines is defined as a 'test point disjunction' or Spider. Spiders may be labeled.

To differentiate these from common Euler diagram notation (normally used to represent set theory) the curves are drawn using dotted and dashed lines.

2.2 PLD Definition

In English: The elements that can be found in a PLD diagram are a number of contours, a number of test points and joining lines that connect test points.

Definition: 1. *A concrete PLD d is a set comprising of a set of closed curves $C = C(d)$, a set of test points $T = T(d)$ and a set of test point joining lines $J = J(d)$.*

$$d = \{C, T, J\}$$

In English: Each element of the diagram has a unique label within the diagram.

Definition: 2. *A minimal region of concrete PLD diagram d is a connected component of*

$$\mathbb{R}^2 - \bigcup_{\hat{c} \in \hat{C}(\hat{d})} \hat{c}$$

Definition: 3. *Let d be a PLD and $\mathcal{X} \subseteq \hat{C}(\hat{d})$ a set of countours. If the set*

$$\hat{z} = \bigcap_{c \in \mathcal{X}} interior(\hat{c}) \cup \bigcap_{\hat{c} \in \hat{C} - X} exterior(\hat{c})$$

is non empty, then $\hat{z}$ is a concrete zone of $\hat{d}$. A zone is a union of minimal regions. The set of all concrete zones of $\hat{d}$ is denoted $\hat{\mathcal{Z}}$.

Each minimal region in the plane may be inhabited by one or more 'test points'. Each test point can be associated with the set of contours that enclose it.

Definition: 4. *$\mathcal{Z}_d : T(d) \to \mathcal{C}$ is a function associating a testpoint with a set of contours in the plane. This corresponds to the interior of the contours defining the zone.*

Pairs of test points may be joined by joining lines. The operator $\overset{join}{\leftrightarrow}$ is used to show that two points are joined by a line in the concrete diagram.

Definition: 5. *$\mathcal{F}_j$ is a function associating a joining line with a pair of test points. The Join t1,t2 is defined as*

$$\mathcal{F}_d : J(d) \to \{t1, t2 \mid t1 \in T(d) \wedge t2 \in T(d) \wedge t1 \neq t2\}$$

In English: Test points on the concrete diagram pair-wise connected by a 'joining line'

A collection of test points connected by joining lines, is an Fuctionally Merged Group, FMG or 'test point disjunction'. An FMG has members which are test points.

may be merged and create a

Definition: 6. *Let d be a PLD : An FMG is a maximal set of test points in d where the test points belong to a sequence connected by joining lines such that:*

$$t_i \overset{join}{\leftrightarrow} t_n, for\ i = 1, ..., n$$

OR consider an FMG as a tree whose nodes are test points.
A singleton test point can be considered a sequence of one test point and is therefore also an FMG.

2.3 Semantics of PLD

- A closed curve in a PLD represents a condition (logical state) being modelled.
- A test point represents the conjunction of the conditions represented by the curves that enclose it.
- A FMG represents the disjunction of all test points that are members of it.

To obtain the set of propositions from a PLD, each FMG must be processed. For each test case in the FMG a new section of the equation is disjuctively appended to it.

Let conjunctive logic equation associated with a test point be determined from the contours that enclose it. i.e. the contours $\mathcal{X}$ from the zone it inhabits.

Definition: 7. *Let $\mathcal{F}_t$ be a function mapping a test point to a proposition / logical equation $p \in P$. The test point inhabits the zone $\mathcal{Z}$ which is a collection of contours (the contours that enclose the test point.*

$$\mathcal{F} : T \rightarrow P$$

$$\mathcal{F}(t) : p = \bigwedge_{c \in \mathcal{Z}} c$$

In English: Thus a 'test point' enclosed by contours labelled a, b, c would be represented by the logic equation $a \wedge b \wedge c$.

Definition: 8. *Let $\mathcal{G}_{fmg}$ be a function that returns a logic equation for a given FMG fmg in the diagram, where an FMG is a non empty set of test points*

$$\mathcal{G} : FMG \rightarrow P_{fmg}$$

The logic equation representing an FMG p_{fmg} can be determined thus.

$$\mathcal{G}_{fmg}(fmg) = \bigvee_{t \in fmg} (\mathcal{F}_t(t))$$

The abstract PLD diagram is a set of logic equations representing all FMGs, along with unused zones (i.e. zones that are not inhabited by FMGs).

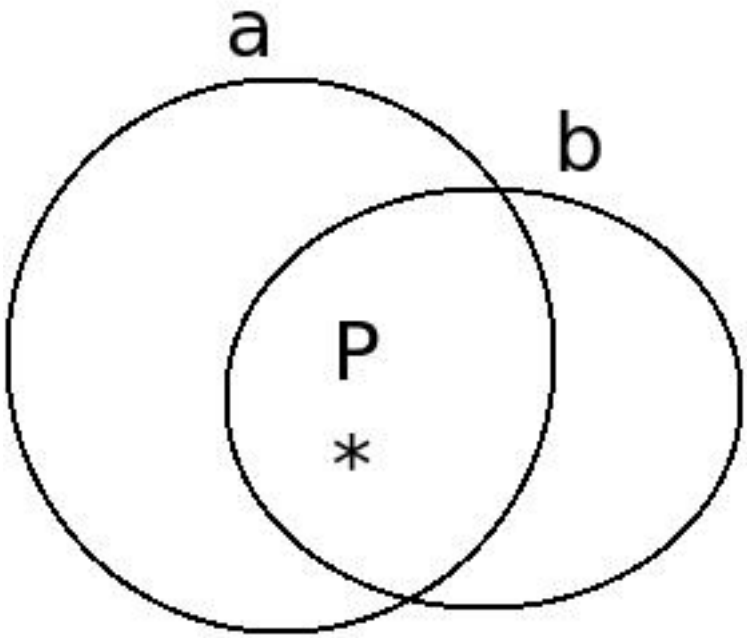
Definition: 9. *A diagram can be reduced to a collection of FMGs. A new diagram can be derived from this, replacing a contour for each FMG. This diagram is at one higher level of abstraction then the diagram that it was produced from.*

3 Example Diagrams

3.1 How to read a PLD diagram

PLD diagrams are read by first looking at the test case points. The test case asterisk will be enclosed by one or more contours. These contours are collected and form the logical conjunction equation for the test case. These test case points thus represent the conjunctive aspects of an equation defined in a PLD. Where these test cases are joined by lines; these represent disjunction of the conjunctive aspects defined by the test cases. Joining lines thus represent dis-junction in a PLD.

3.2 Logical AND example



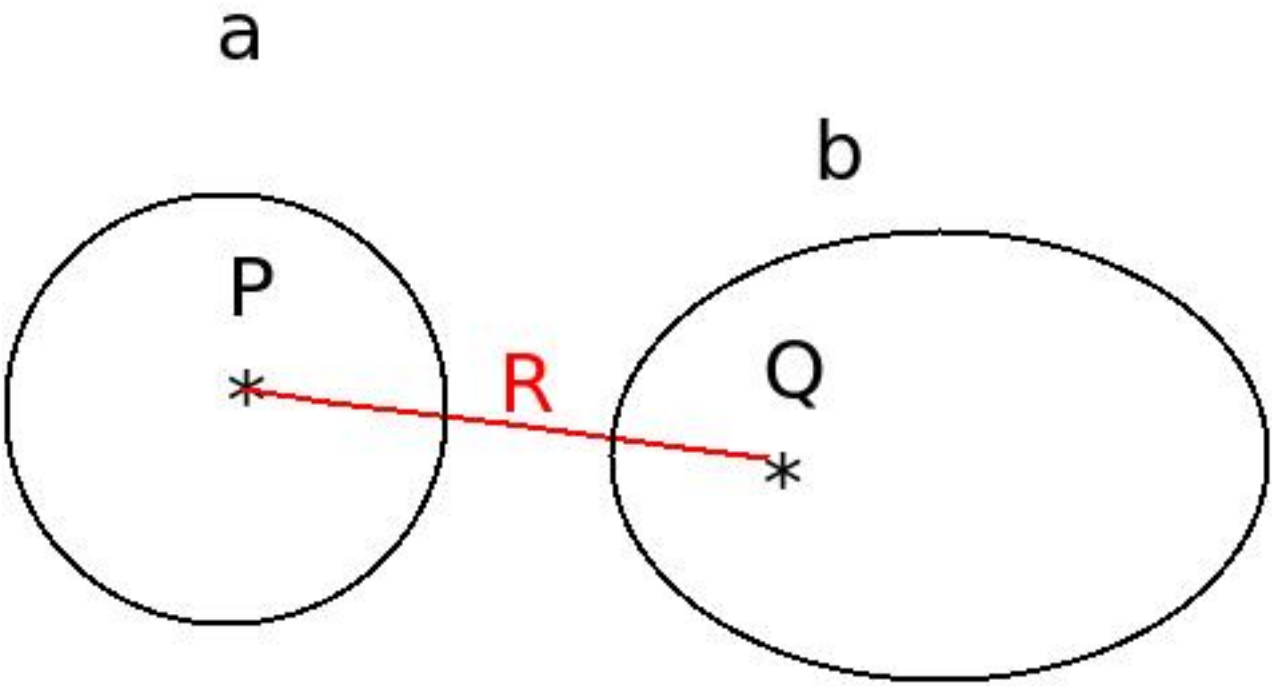
In the diagram 3.2 the area of intersection between the contours a and b represents the conjunction of those conditions. The point P represents the logic equation

$$P = (a \wedge b)$$

There are no disjunctive joining lines and so this diagram represents one equation only, $P = (a \wedge b)$.

How this would be interpreted in failure analysis In failure analysis, this could be considered to be a sub-system with two failure states a and b. The proposition P considers the scenario where both failure modes are active.

3.3 Logical OR example



The diagram **??** is converted to Boolean logic by first looking at the test cases, and the contours they are placed on.

$$P = (a)$$

$$Q = (b)$$

The two test cases are joined by a the line named R. we thus apply disjunction to the test cases.

$$R = P \vee Q$$

substituting the test cases for their Boolean logic equations gives

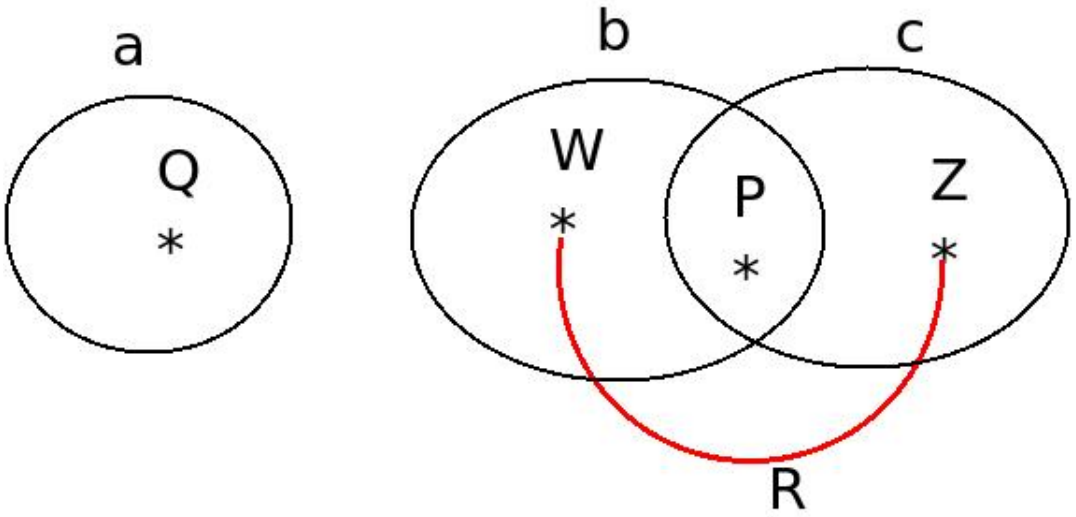
$$R = ((a) \vee (b))$$

.

3.4 Labels and useage

In diagram **??** Z and W were labeled but were not necessary for the final expression of $R = b \vee c$. The intended use of these diagrams, is that resultant logical conditions be used in a later stage of reasoning. Test cases joined by disjunction, all become represented in one, resultant equation. Therefore only test cases not linked by any disjunctive joining lines need be named.

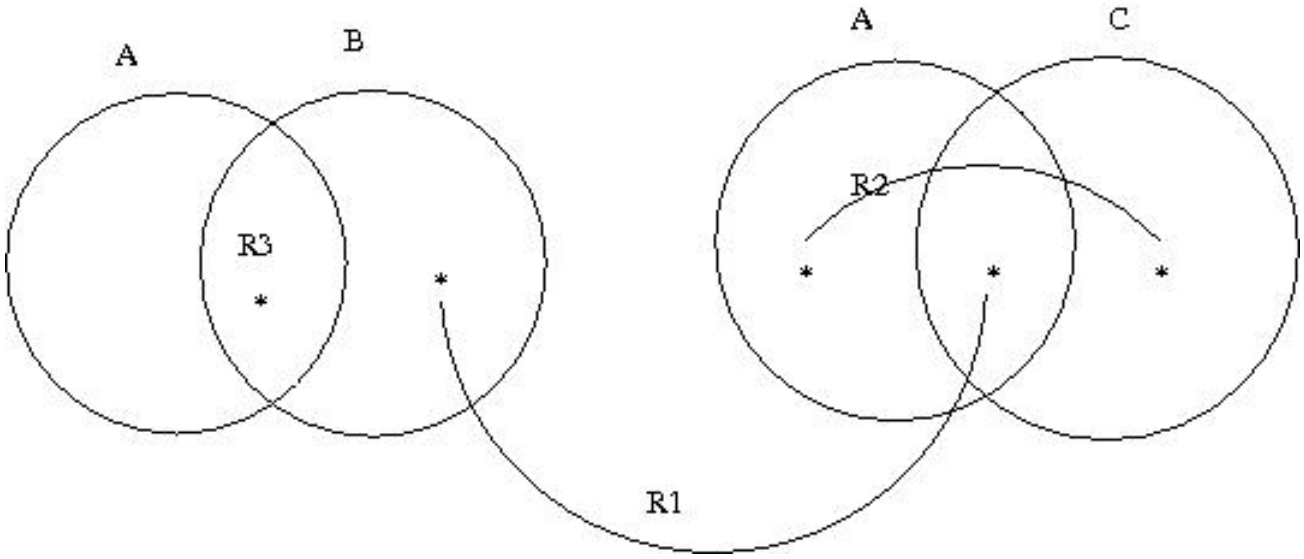
The diagram **??** can therefore be represented as in diagram **??**, with two unnamed test cases.



How this would be interpreted in failure analysis In failure analysis, this could be considered to be a sub-system with two failure states a and b. The proposition P considers the scenario where either failure mode is active. Additionally it says that either failure mode a or b being active will have a resultant effect R on the sub-system. Note that the effect of a and b both being active is not defined on this diagram.

3.5 Repeated Contour example

Repeated contours are allowed in PLD diagrams. Logical contradictions or tautologies can be detected automatically by a software tool which assists in drawing these diagrams.



The diagram **??** is converted to Boolean logic by first looking at the test cases, and the contours they are placed on.

$$P = (b)$$

$$Q = (a) \wedge (c)$$

The two test cases are joined by a the line named $R1$. we thus apply disjunction to the test cases.

$$R1 = P \vee Q$$

$$R1 = b \vee (a \wedge c)$$

.

$R2$ joins two other test cases

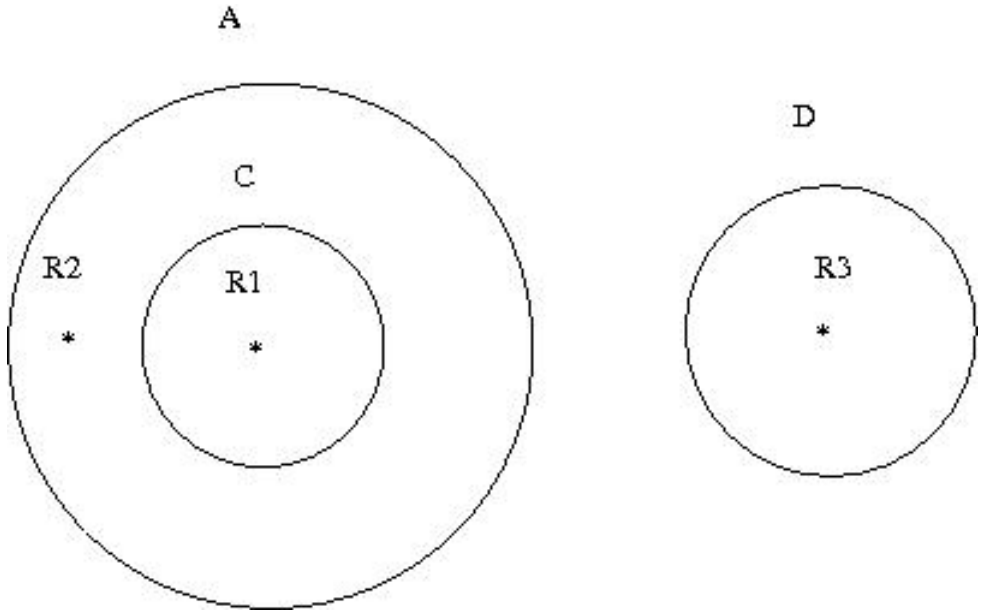
$$R2 = a \vee c$$

The test case residing in the intersection of countours B and A represents the logic equation $R3 = a \wedge b$.

How this would be interpreted in failure analysis In failure analysis, $R2$ is the symptom of either failure mode A or C occurring. $R1$ is the symptom of B or $A \wedge C$ occurring. There is an additional symptom, that of the test case in $A \wedge B$.

3.6 Inhibit Failure

Very often a failure mode can only occurr given a searate environmental condition. In Fault Tree Analysis (FTA) this is represented by an inhibit gate.



The diagram **??** has a test case in the contour C. Contour C is enclosed by contour A. This says that for failure mode C to occur failure mode A must have occurred. A well known example of this is the space shuttle 'O' ring failure that caused the 1986 challenger disaster [**?**]. For the failure mode to occurr the ambiant temperature had to be below a critical value. If we take the failure mode of the 'O' ring to be C and the temperature below critical to be A, we can see that the low temperature failure mode C can only occurr if A is true. The 'O' ring could fail in a different way independant of the critical temperature and this is represented, for the sake of this example, by contour D.

In terms of propositional logic, the inhibit gate of FTA, and the countour enclosure of PLD represent *implication*.

c	a	$R1$
F	F	T
F	T	T
T	F	F
T	T	T

$$R1 = c \implies a$$
$$R2 = a$$
$$R3 = d$$

How this would be interpreted in failure analysis In failure analysis, $R2$ is the symptom of either failure mode A or C occurring. $R1$ is the symptom of B or $A \wedge C$ occurring. Note that although R2 is a symptom of the sub-system, on its own it will not lead to a dangerous failure mode of the subsystem.

4 Intended use in FMMD

The intention for these diagrams is that they are used to collect component faults and combinations thereof, into faults that, at the module level have the same symptoms.

4.1 Example Sub-system

For instance were a 'power supply' being analysed there could be several individual component faults or combinations that lead to a situation where there is no power. This can be described as a state of the powersupply being modeelled as NO_POWER. These can all be collected by DISJUCNTION, i.e. that this this or this fault occuring will cause the NO_POWER fault. Visually this disjuction is indicated by the joining lines. As far as the user of the 'power supply' is concerned, the power supply has failed with the failure mode NO_POWER. The 'power supply' module, after this process will have a defined set of fault modes and may be considered as a component at a higher level of abstraction. This module can then be combined with others at the same abstraction level. Note that because this is a fault collection process the number of component faults for a module must be less than or equal to the sum of the number of component faults.

`CVS Revision Identity $Id: logic_diagram.tex,v 1.17 2010/01/06 13:41:32 robin Exp $`

Compiled last January 9, 2010

`$Id: paper.tex,v 1.4 2009/11/28 20:05:52 robin Exp $`