SELBSTDIAGNOSE VON MEHRRECHNERSYSTEMEN

BEI NICHTVOLLSTÄNDIGER VERNETZUNG

D i s s e r t a t i o n

zur Erlangung des Grades eines Doktors
der Naturwissenschaften

der Fakultät für Physik
der Eberhard-Karls-Universität zu Tübingen

vorgelegt von
Rüdiger W. Brause
aus Borna
1983

Tag der mündlichen Prüfung: 19.Dezember 1983

Dekan: Prof. Herrmann
1.Berichterstatter: Prof.Dr.M.Dal Cin
2.Berichterstatter: Prof.Dr.W.Güttinger

Abstract

BRAUSE, RÜDIGER:
SELBSTDIAGNOSE VON MEHRRECHNERSYSTEMEN BEI NICHT VOLLSTÄNDIGER VERNETZUNG

Ausgehend von den graphentheoretischen und den wahrscheinlichkeits- theoretischen Modellen der Diagnose von Mehrrechnersystemen wird die Diagnose als Mustererkennungsproblem aufgefaßt und mit den entsprechenden Methoden behandelt. Die daraus resultierende Bayes-Diagnose reduziert sich bei einem 0-1 Loss zu einer Wahrscheinlichkeits-Diagnose. Nach einem Vergleich mit der aus den graphentheoretischen Modellen resultierenden deterministischen Diagnose werden die Bedingungen genauer betrachtet, unter denen eine Diagnose in einem Mehrrechnersystem abläuft.

Dazu wird untersucht, unter welchen Bedingungen eine Bayes-Diagnose in einem vernetzten Mehrrechner-System abläuft, in dem dezentral von jeder beteiligten Einheit getestet und diagnostiziert wird. Da es nicht immer möglich ist, zur Lokalisierung eines Defekts das gesamte Mehrrechner-System abzuschalten, wird ein Fehlermodell und ein Diagnoseverfahren beschrieben, die unter der Voraussetzung einer begrenzten Fehlerfortpflanzung (nicht vollständige Vernetzung des Mehrrechnersystems) für Tests und Diagnose nur einen begrenzten Teil des Gesamtsystems benutzt.

Die Verwendung dieses Diagnoseverfahrens bei Mehrrechner-Systemen in VLSI-Realisierung ('systolische Felder') wird diskutiert und es werden Bedingungen angegeben, unter denen sich ein solches VLSI-Rechnernetz unter Zuschaltung von redundanten Einheiten nach der Diagnose auch selbst reparieren kann.

Inhalt

1. Einleitung

Unsere heutige Zeit ist geprägt durch eine rasante technologische Entwicklung auf dem Gebiet der Mikroelektronik. Kaum 10 Jahre nach Einführung des ersten 4-Bit Mikroprozessors werden schon die ersten 32-Bit Mikroprozessoren mit einer Komplexität von 450000 Transistorfunktionen präsentiert. Zwar lassen sich mit einer derartigen Anhäufung von Funktionen auf kleinstem Raum sehr komplizierte Aufgaben verwirklichen, die vorher mit vielen Platinen, unzuverlässigen Steckkontakten etc. realisiert waren, jedoch ergeben sich aus dieser Höchstintegration neue Probleme:
Wie soll man ein solch kompliziertes Gebilde bei der Fertigung und später im Betrieb auf Funktionsfähigkeit testen?
Zusätzlich verlangen die neuen Anwendungen dieser preiswerten, kleinen und doch leistungsfähigen Systeme in der Luft- und Raumfahrt, in der Medizintechnik und in anderen sicherheitskritischen Bereichen wie Verkehrskontrolle und Atomreaktoren, daß die technischen Systeme - trotz wachsender Komplexität und damit höherer Ausfallwahrscheinlichkeit - ausfallsicherer werden. Dies führte zu verstärkten Bemühungen in den 70-er Jahren, Entwurfsfehler bereits im Ansatz zu vermeiden durch systematische, genaue Schaltungsentwicklung (Hardware) mit Hilfe von CAD und CAM und durch Programmentwicklung (Software) mit Methoden der strukturierten Programmierung und der rechnergestützten Validierung und Dokumentation. Andererseits wurden solche Systeme erforscht, entwickelt und gebaut, die beim Betrieb auftretende Fehler rechtzeitig erkennen und diese ohne Eingriffe des Benutzers selbsttätig behandeln.
Die vorliegende Arbeit läßt sich in die Bemühungen der zweiten Gruppe einreihen. In Kapitel 2 und 3 werden dazu die wichtigsten Modelle zur Fehlerdiagnose vorgestellt. Um den so unterschiedlichen, komplizierten Testmethoden mit Oszilloskop, Logikanalysator und Testsoftware gerecht zu werden, beschränken sich die Modellannahmen über die Tests weitgehend auf die entscheidende Aussage jedes Tests, ob die getestete Elektronik defekt ist oder nicht. Entsprechend wird auch das Gesamtsystem in testbare Einheiten mit statistisch unabhängigen Lebensdauern aufgeteilt, wobei die Einheiten nur zwei Zustände haben können : intakt oder defekt. Je nach Modellannahmen können Einheiten auch für den Test einer anderen Einheit eingesetzt werden.
Dieser Sachverhalt läßt sich durch einen gerichteten Graphen $G(V,E)$ mit der Knotenmenge V und der Kantenmenge E beschreiben. Die Knoten werden dabei den

Einheiten u_i und die Kanten den Testverbindungen im System zugeordnet. Der gerichtete Graph wird 'Testgraph' genannt; die Testergebnisse sind somit Gewichtungen der Kanten. Ein Beispiel ist in Abb.1a zu sehen.

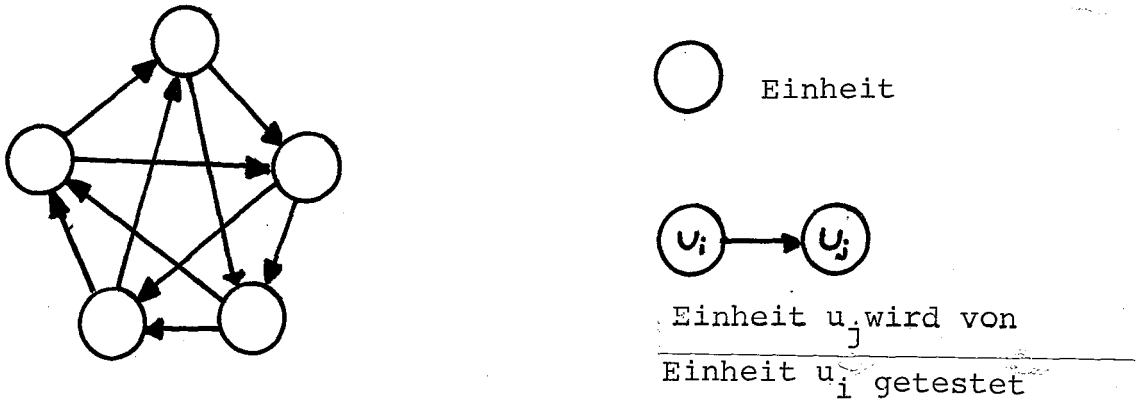


Abb.1a Testgraph mit N=5 Einheiten

Die geordnete Menge S der |E| Testergebnisse wird als 'Syndrom' bezeichnet. Das Grundproblem der Diagnose besteht darin, bei gegebenem Testgraphen und gegebenem Syndrom herauszufinden, welche Einheiten defekt sind. Dies läßt sich in folgende Unterprobleme aufgliedern:

-- Finde heraus, unter welchen Umständen das Problem lösbar ist.

-- Finde optimale Testgraphen, die mit möglichst wenigen Tests möglichst viele Fehler lokalisieren.

-- Finde Algorithmen für die Diagnose.

Als erste gingen Preparata et alii in ihrer klassischen Arbeit /PRE/ die zwei erstgenannten Probleme an. Sie bezeichnen ein System als '1-Schritt t-Fehler diagnostizierbar' oder auch 't-diagnostizierbar ohne Reparatur', wenn mit der Kenntnis des Testgraphen und des Syndroms alle fehlerhaften Einheiten sofort lokalisiert werden können, vorausgesetzt, es sind nicht mehr als t Einheiten defekt. Dagegen wird ein System 'sequentiell t-Fehler diagnostizierbar' oder auch 't-diagnostizierbar mit Reparatur' genannt, wenn von maximal t fehlerhaften Einheiten des defekten Zustands mindestens eine Einheit identifiziert werden kann.

Die Begriffe 'Test' und 'Einheit' sind sehr allgemein aufzufassen. Betrachten wir dazu als Beispiel das fehlertolerante Multi-Mikroprozessorsystem ATTEMPTO, bei dem der Verfasser mitarbeitet /ATT/. ATTEMPTO besteht aus einer Zahl von unabhängigen Single-Board Computern, die durch zwei Busse miteinander gekoppelt sind: Ein 'Terminal- Bus' dient zur Kommunikation zwischen dem Benutzer und den Computern und ein Kommunikationsbus zum Nachrichtenaustausch und damit auch zur Synchronisierung der Computer.

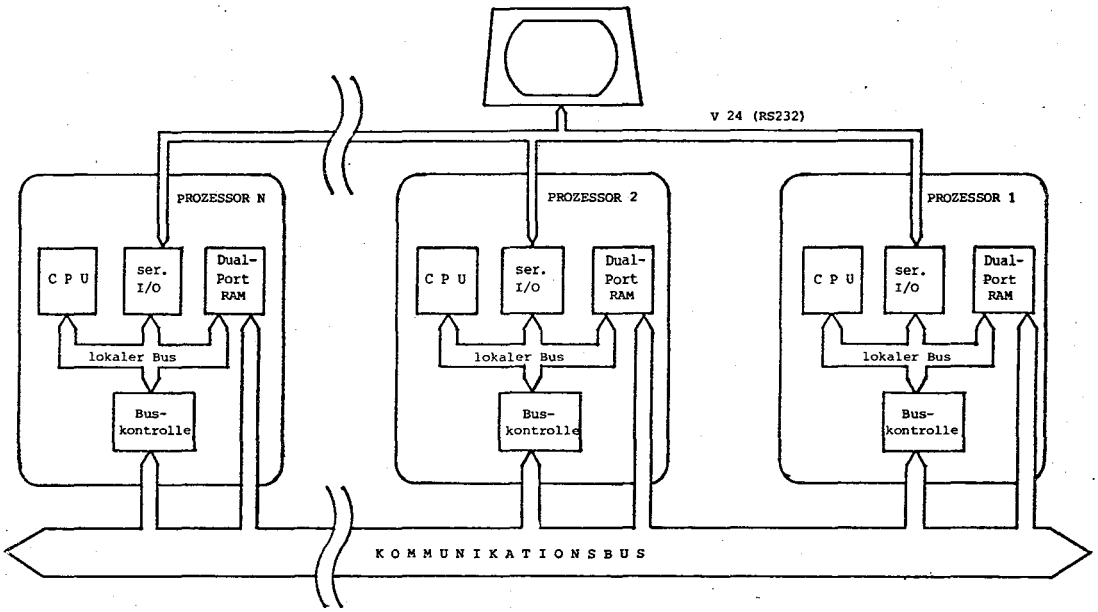


Abb 1b Der fehlertolerante Rechner ATTEMPTO

Bei fehlertolerantem Betrieb arbeiten mehrere der Single-Board Computer parallel Kopien des gleichen Programms ab und synchronisieren sich bei der Ausgabe.

Mit dem Begriff 'Einheit' kann man in diesem System jeden Single-Board-Computer bezeichnen und mit dem Begriff 'Test' eine Operation, bei der die komprimierten Ausgabedaten (Signaturen) der verschiedenen Computer auf Übereinstimmung verglichen werden.

Die Ergebnisse dieser Vergleichsoperationen werden von jedem einzelnen Computer diagnostiziert. Bei der Diagnose der Ergebnisse solcher 'Vergleichstests' findet das Modell aus Kapitel 2.3 Verwendung. Danach wird durch Nachrichtenaustausch derjenige Computer aus der Menge der Intakten bestimmt, der die korrekten Ausgabedaten ausgibt.

In ATTEMPTO bestehen somit die Testdaten primär aus den normalen Benutzerdaten; die Tests sind 'funktionsbegleitend'. Dies ist ein wichtiger Vorteil bei

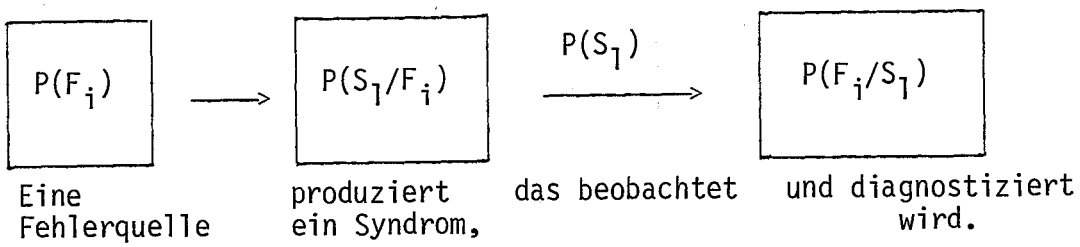
zeitkritischen Programmen, die eine Unterbrechung zum Testen des Rechners nicht erlauben. Außerdem sind für die Leerlaufzeiten (Warten, etc) der Rechner besondere Selbsttestprogramme vorgesehen, um eine unbemerkte Ansammlung von Fehlern in Rechner-Instruktionen, Speicher oder anderer Hardware zu vermeiden, die vom Benutzerprogramm nicht verwendet werden.

Das von Preparata et alii entwickelte Diagnosemodell ist in bestimmten Situationen nicht anwendbar, ebenso wie die anderen graphentheoretischen Modelle, die in Kapitel 2 referiert werden. Dies soll näher erläutert werden. Eine geläufige Annahme der Modelle besteht darin, daß beim Einsatz von intakten Einheiten immer korrekte Testergebnisse erzielt werden (Vollständigkeitsannahme der Tests). Leider ist diese Annahme nicht immer zutreffend. Bei hochkomplexen Bauelementen können die Tests meist nur grob die wichtigsten Funktionen testen, da ein systematisches Überführen des Bauelements in alle möglichen Zustände wegen der Komplexität der Bauelemente in einer vertretbaren Testzeit nicht möglich ist. Das Testergebnis kann somit nur noch eine Wahrscheinlichkeitsaussage über den Zustand des Bauelements darstellen, selbst wenn die testende Einheit intakt ist.

Damit sind die graphentheoretischen Modelle nicht mehr anwendbar. Es wird stattdessen ein Diagnoseverfahren benötigt, bei welchem von vorliegenden Testergebnissen mit bestimmter Wahrscheinlichkeit auf den Systemzustand geschlossen werden kann.

Diese Anforderung wird durch das Verfahren der stochastischen Mustererkennung erfüllt. Hierbei wird von einem beobachteten Muster mit bestimmter Wahrscheinlichkeit auf eine mustererzeugende Klasse geschlossen. In Kapitel 4 wird deshalb der Ansatz der Mustererkennung referiert und auf die Fragestellung angewendet, wie mit dem kleinsten Risiko jedem Syndrom (Menge der Testergebnisse) eine Fehlerklasse (Menge der defekten Einheiten) zugeordnet werden kann.

Betrachten wir die Fragestellung der stochastischen Mustererkennung genauer:
Sei $P(F_i)$ die a-priori Wahrscheinlichkeit für das Auftreten einer Fehlerklasse F_i. Diese Fehlerklasse erzeugt mit der bedingten Wahrscheinlichkeit $P(S_l/F_i)$ das Syndrom S_l, das mit der Wahrscheinlichkeit $P(S_l)$ auftritt.
Wenn nun die a-posteriori Wahrscheinlichkeit $P(F_i/S_l)$ für F_i bekannt wäre, so ließe sich von dem auftretenden Syndrom S_l auf die verursachende Fehlerklasse F_i schließen.



Werden noch zusätzliche Kosten eingeführt, die bei einer falschen Diagnose entstehen, so gibt es, wie in Kapitel 4 gezeigt wird, eine für diese Problemstellung optimale Diagnose, die sog. Bayes-Diagnose. Sie stellt eine sehr allgemeine Diagnose dar, die, wie in Kapitel 5 beschrieben, mit den speziellen Annahmen der graphentheoretischen Diagnosemodelle aus Kapitel 2 und 3 für $P(S_1/F_i)$ und bekannten Ausfallwahrscheinlichkeiten gleich gute oder bessere Ergebnisse liefert als jene. Außerdem ist eine Diagnose im Unterschied zu den graphentheoretischen Modellen auch bei mehr als t defekten Einheiten möglich.
Den Vorteilen der Bayesdiagnose stehen aber auch mehrere Nachteile entgegen. Zum einen werden für diese Diagnose mehr Informationen (Ausfallwahrscheinlichkeiten der Einheiten, etc.) benötigt, die nicht einfach zu bestimmen sind /BRA1/. Zum anderen ist die Laufzeit-Komplexität des Diagnosealgorithmus wesentlich höher als bei den deterministischen Diagnosen der graphentheoretischen Modelle (s.Anhang A). Außerdem ist es sehr schwierig, für die Bayes-Diagnose gute Testgraphen zu konstruieren.

Die bisher geschilderten Diagnosemodelle beschränken sich darauf, bei bekannter Systemstruktur und vorliegenden Testergebnissen Diagnoseverfahren anzugeben. Ein wichtiges Problem stellt dabei die Frage dar, wie solche Diagnoseverfahren in einem Computersystem organisatorisch eingesetzt werden können.
In den meisten Anwendungsfällen handelt es sich bei den Einheiten, die von den Modellen betrachtet werden, um Computersysteme, die von (meist) qualifiziertem Service-Personal gewartet werden. Dies ist aber sehr kostspielig und in manchen Fällen (Satelliten) nicht möglich. Es liegt deshalb nahe, auch für die Diagnose einen Computer zu verwenden. Die Auswertung aller Syndrome, die Reparatur und die Rekonfiguration des Systems geschieht somit von einem zentralen, übergeordneten Computer, der aber selbst nicht defekt sein darf ('hard-core'). Dieses Diagnoseverfahren wird bei vielen existierenden Computersystemen verwendet, z.B. Siemens SMS 201, Plessey 250 und wird auch

für die Kategorie der SMD (Single Instruction, Multiple Data) Maschinen verwendet, bei denen eine Vielzahl von 'processing elements' nur einer einzigen Kontrolle unterworfen sind, wie z.B. bei der ILLIAC IV (s./MIE/).
Im Gegensatz dazu bieten Systeme von gekoppelten, unabhängigen Rechnern, Maschinen der Kategorie MIMD (Multiple Instruction, Multiple Data), die Möglichkeit, zusätzlich zu dem Test von Rechner zu Rechner ('dezentrales Testen') auch die Diagnose dezentral von den noch intakten Rechnern durchzuführen. Damit entfällt die Notwendigkeit einer besonders ausfallsicheren Einheit und es wird möglich, durch Beschränkung des Testens und Reparierens auf Teile des Systems (Computerverbunds), den Rechnerbetrieb bei reduzierter Gesamtsystemleistung aufrecht zu erhalten. Derartige Computernetze sind zur Zeit als Verbund bestehender, autonomer Rechenanlagen zur Nutzung gemeinsamer, meist kostenträchtiger Resourcen (Plattenspeicher, Schnelldrucker, Datenbanken) üblich ('In-house networks') /PHI/,/INF/. In Abb.1c ist ein solches Netzwerk nach den Vorstellungen der Fa.Zilog abgebildet.

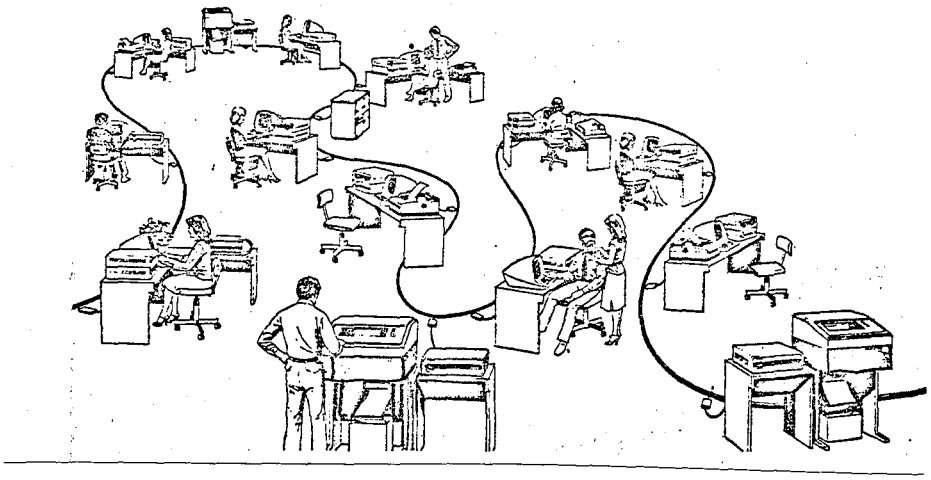


Abb. 1c In-house network

Für eine Anwendung in Computernetzen müssen die graphen- und wahrscheinlichkeitstheoretischen Diagnose-Verfahren erst in dezentrale Verfahren umgesetzt werden. Solche dezentrale Verfahren ergeben sich aber nicht sofort aus den vorher beschriebenen, zentralistischen Modellen. Viele Verfahren (Markoffmodelle etc.)lassen sich nur schwer auf dezentrale, asynchrone Entscheidungen übertragen /TEN/. Vielmehr muß bei einzelnen, autonomen Einheiten der gesamte Ansatz neu formuliert werden.
Ein vollständig dezentraler Diagnosealgorithmus wurde zuerst von Kuhl und Reddy /KUH1/ vorgestellt. Darin wird angenommen, daß die Lebensdauern der Einheiten

statistisch unabhängig sind. Die Einheiten sollen nur über den Austausch von Nachrichten miteinander gekoppelt sein. Der Diagnosealgorithmus ist aus der Sicht einer einzelnen Einheit formuliert und gibt an, was die betreffende Einheit bei Erhalt einer Nachricht über durchgeführte Tests tun soll. Dabei übernimmt jede Einheit nur Testergebnisse derjenigen Einheiten, die als 'intakt' angesehen werden. Dies sind zuallererst die direkten Nachbarn, die von jeder Einheit getestet werden, dann deren intakte Nachbarn und alle weiteren Nachbarn, die als intakt bekannt sind. Nachrichten von Nachbarn, die als 'defekt' angesehen werden, werden ignoriert.
Unter der Annahme der Vollständigkeit der Tests läßt sich zeigen, daß der Diagnose-Algorithmus immer eine korrekte Diagnose bewirkt, vorausgesetzt, es sind nicht mehr als t Einheiten ausgefallen. Diese obere Schranke t, bei der noch eine korrekte Diagnose möglich ist, wird hier durch den Knotenzusammenhang $k(G_V)$ des gerichteten Verbindungsgraphen G_V gebildet, das heißt durch die kleinste Zahl von Einheiten, deren Ausfall das Netz von gekoppelten Rechnern zerfallen läßt.
Wenn die Annahme der Vollständigkeit der Tests nicht mehr zutrifft, so ist es für eine intakte Einheit auch nach dem Testen der Nachbarn nicht eindeutig, welche Nachbarn defekt sind. Damit ist das obige dezentrale Diagnoseverfahren nicht mehr anwendbar. Stattdessen müssen alle Testergebnisse gesammelt werden, die von den Nachbarn übermittelt werden. Defekte Einheiten können aber die Testergebnisse anderer Einheiten beim Übermitteln ändern. Um Fälschungen zu vermeiden, ist es üblich, die Nachricht zu verschlüsseln; in vielen Rechnernetzen (Banken, Militär) ist eine Verschlüsselung schon aus Sicherheitsgründen notwendig. In Kapitel 6.2 wird ein Algorithmus entwickelt, der in Rechnernetzen bei weniger als $k(G_V)$ defekten Einheiten eine Bayesdiagnose dezentral ausführt, ohne vollständige Tests vorauszusetzen. Er bewirkt, daß jede intakte Einheit die gleiche Sicht des Gesamtsystems hat. Sind die Tests zudem vollständig, so ist die Sicht sicher korrekt.
Welche Struktur haben nun die Rechnernetze, für die eine dezentrale Diagnose geeignet ist?
Unter den Verbindungsstrukturen von Computernetzen nehmen die regulären Netze einen besonderen Platz ein, da sie sich durch ihre regelmäßigen Strukturen von Einheiten gleicher Bauart besonders für die Herstellung in VLSI- Technologie eignen. Die Herstellung der für die Produktion notwendigen fotochemischen oder elektronischen Ätzmasken sowie die Produktionsprozesse, -kontrolle und Bauteiletests werden dadurch stark vereinfacht, vereinheitlicht und somit deutlich billiger. Dieser Kostenvorteil wirkt sich auch bei der Softwareerstellung stark aus (vgl. dazu die Erfahrungen im C.mmp- System

/JON/).
Außerdem wird eine automatische Diagnose umso interessanter, je weniger man in ein System von außen eingreifen kann. Für höchstintegrierte Chips mit mehreren Mikrocomputern auf einem Chip sind gute Test- und Diagnosestrategien deshalb von großer Bedeutung.
Eine durch die VLSI-Technologie gegebene Nebenbedingung ist die Forderung, das gesamte Multicomputernetz als Datenmaschine auf der Fläche eines Siliziumscheibchens zu realisieren. Als Kommunikations- (und damit auch als Teststrukturen) eignen sich deshalb besonders kreuzungsfreie Flächennetze, bei denen die Ebene aus gleichmäßigen, identischen Flächeneelementen aufgebaut ist. In Kapitel 7 wird gezeigt, daß dies nur mit Dreiecken, Vierecken oder Sechsecken möglich ist. Die folgende Abbildung 1.d zeigt die daraus resultierenden Netztypen.

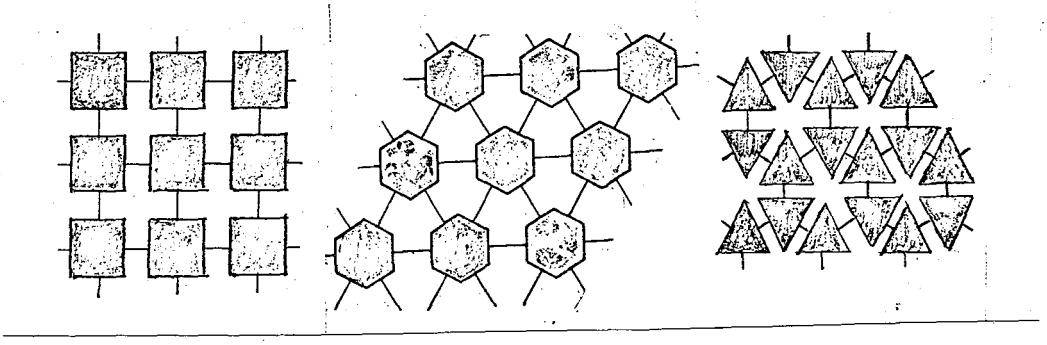
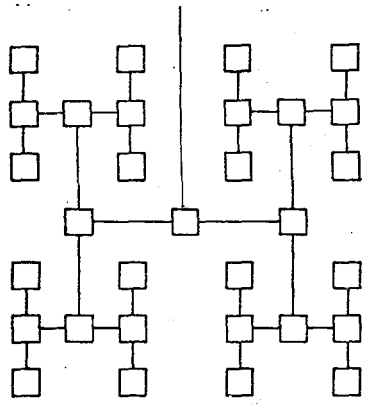
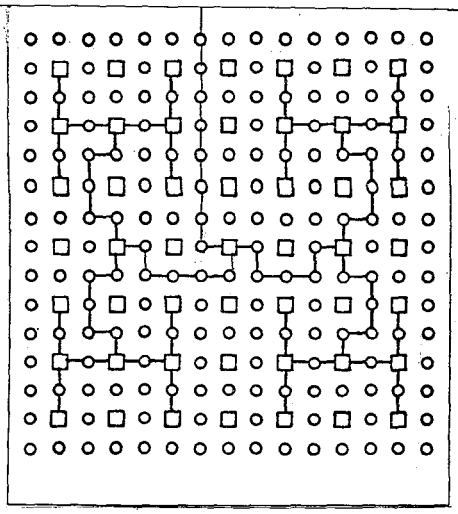


Abb. 1d Typen von VLSI- Netzen

Für zwei dieser Netztypen gibt es in der Literatur schon Anwendungen, beispielsweise unter der Bezeichnung 'Holland-Rechner' und 'systolische Felder'. Sie sind für schnelle Matrizenmultiplikation vorgesehen. Mit der Integration von zusätzlichen Schaltern zur anwendungsbezogenen Konfiguration der Rechner lassen sich daraus aber auch universelle Multicomputersysteme realisieren. In Abb.1e ist die Struktur eines Baum-Rechners zu sehen, wie er in VLSI-Ausführung für Suchprobleme vorgeschlagen wurde /BEN2/. Abbildung 1f zeigt die Realisierung dieses Baums auf einem universellen Multicomputer; die Kreise bezeichnen Schalter und die Vierecke die Prozessoren (aus /SNY/).



e)



f)

Abb 1e,f Realisierung eines Baumrechners

In den bisher geschilderten, dezentralen Diagnoseverfahren wurde jeweils die Existenz eines Syndroms vorausgesetzt, das durch Testen des gesamten Systems erlangt wurde. Die dazu benötigten Tests werden gleichzeitig oder hintereinander ausgeführt, ohne daß dabei Benutzerprogramme laufen, wie es beispielsweise beim Anschalten und Initialisieren eines Systems geschehen kann.

Wenn dem Multicomputersystem lebenswichtige Aufgaben übertragen werden (Flugsicherung, Verkehrskontrolle, etc.), ist aber eine vollständige Blockierung des Systems durch Testen nicht tragbar. Eine Alternative besteht darin, die Test-tasks sukzessive auf verschiedenen Einheiten und parallel zum normalen Betrieb im System abarbeiten zu lassen. Nach Durchführung des allerletzten Tests wird dann die Diagnose vorgenommen. Dies vermeidet zwar die Blockierung des Systems, schafft aber neue Probleme, da ein defekter Prozessor vor seiner Reparatur fehlerhafte Daten erstellen und weitergeben kann. Eine Korrektur dieser Fehler nach der Reparatur ist kaum möglich (vgl./JON/) und läßt nur eine Initialisierung des Gesamtnetzes zu.
Eine andere Möglichkeit besteht darin, Mechanismen zur Fehlererkennungs derart in das System einzubauen, daß auftretende Fehler bereits im Normalbetrieb erkannt (funktionsbegleitende Tests) und dann weitere Tests eingeleitet werden können.

Da eine größere Zahl von miteinander verbundenen Rechnern meist nicht vollständig vernetzt ist (Leitungsprobleme), bleibt die Kommunikation und damit die Fehlerfortpflanzung beschränkt. Man kann nun versuchen, anstelle des gesamten Netzes nur den Bereich im Netz zu testen und zu diagnostizieren, in dem ein Fehler auftritt ('lokale Diagnose'). Abgesehen von der geringeren Testbelastung des Systems ist auch durch die kleinere Zahl von Einheiten und Tests die Diagnosezeit dabei kürzer als bei der Diagnose des gesamten Systems. In solchen Netzen können somit die restlichen Einheiten des Systems die wichtigsten Benutzertasks abarbeiten.
In Kapitel 8.1 wird untersucht, wie eine Fehlererkennung und -lokalisierung parallel zur Bearbeitung einer Aufgabe durch das Rechnernetz stattfinden kann. Das dort vorgestellte Verfahren basiert auf lokalen Tests, die aus der Überprüfung der vom Nachbarn im Normalbetrieb gesendeten Daten auf Formattreue, Konsistenz und Plausibilität bestehen. Ergänzt wird diese Art der Fehlererkennung durch periodische Tests der Nachbarn. Je nach System kann die Diagnose durch einen zentral ausgeführten Algorithmus einer zentralen Service-Einheit (Kapitel 8.2) oder mit einem dezentralen Algorithmus von jeder Einheit (Kapitel 8.3) geschehen.
Es wird sodann durch schrittweises Vergrößern des Testgraphen ('Teststufen') in Form einer 'Testaktivierung' der benachbarten Einheiten die Umgebung der Einheit, die den Fehler erkannt hat, getestet. Der Testgraph wird solange ausgeweitet, bis mit großer Wahrscheinlichkeit feststeht, daß der Defekt erfaßt ist. Anschließend wird der lokale Netzzustand diagnostiziert.
Ein wichtiger Vorteil des beschriebenen Diagnoseverfahrens ist seine konzeptionelle Unabhängigkeit vom Testgraphen und damit von der Struktur des verwendeten Rechnernetzes, so daß bei Änderung der Netzstruktur (Ausfall, Rekonfiguration, Erweiterung um zusätzliche Einheiten) keine Änderung des Algorithmus nötig ist.
Für Systeme wie die regulären Flächennetze aus Abb.1d, die einen kleinen Knotenzusammenhang k(G) (wenige Nachbarn pro Einheit), aber viele Einheiten haben, ist die Voraussetzung von weniger als k(G) defekten Einheiten ziemlich unrealistisch. Zwar werden Teilmengen des Rechnernetzes eine korrekte, dezentrale Diagnose ausführen können, aber defekte Einheiten können parallel dazu ein anderes Diagnose-Ergebnis errechnen. Wenn kein Eingriff in das System von außen möglich ist (Satelliten) oder gewünscht wird (Service-Probleme), ist ein automatisches Verfahren nötig, um herauszufinden, welches Diagnoseergebnis richtig ist.
Da wir nicht annehmen können, daß alle defekte Einheiten ein bestimmtes Verhalten zeigen, können wir nicht a-priori entscheiden, welche Gruppe von

Recheneinheiten das richtige Diagnoseergebnis ermittelt hat. Nehmen wir nun zusätzlich an, daß in dem betrachteten System die Gruppe der intakten Einheiten größer ist als die der defekten Einheiten, so läßt sich das Problem durch eine Mehrheitsentscheidung lösen. Da die intakten Einheiten sich nicht unbedingt mit den defekten Einheiten koordinieren können, muß sich die Mehrheitsentscheidung der Diagnose schließlich in der Reparatur bzw. Isolierung der defekten Einheiten niederschlagen.

Betrachten wir nun ein solches System näher.

Es sei angenommen, daß im System ein Abschalten von Einheiten und ein Zuschalten von Reserveeinheiten möglich ist. Bei manchen Anwendungen muß dabei die Netzstruktur gewahrt bleiben (systolische Felder!). Da es in diesen Netzen nicht sinnvoll ist, einen 'pool' von Ersatzeinheiten anzulegen, die an die Stelle von jeder Einheit schaltbar sind (Leitungsprobleme!), empfiehlt es sich, z.B. jede Einheit redundant mit einer Reserveeinheit (kalte Reserve) auszulegen, die im Fehlerfall an die gleiche Stelle treten kann, ohne vorher im Netz enthalten zu sein.

Außerdem ist für die erwähnten VLSI- Multiprozessorsysteme eine Reparatur defekter Einheiten auch unter dem Blickwinkel der Produktion interessant. Bei der Herstellung von VLSI-Chips ist die Ausbeute meist nicht sehr hoch. Deshalb wird heutzutage z.B. bei der Produktion von Speicherchips ab einer Kapazität von 64KBit dazu übergegangen, redundante Speicherzellen vorzusehen, die in der Fertigung durch Schmelzschalter an die Stelle von defekten Speicherzellen geschaltet werden. Diese Schalter (Diodenstrecken, die durch Stromüberlastung nichtleitend werden, oder Metallbrücken, die mit Laserstrahlen durchtrennt werden), funktionieren aber naturgemäß nur einmal. Die Diagnose- und Reparaturalgorithmen dürfen also immer nur eine korrekte Reparatur erlauben.

VLSI-Chips gestatten meist zwar eine Rekonfiguration des Systems mit Reserve-Einheiten bei der Herstellung oder beim Start-up Test, ermöglichen aber durch die begrenzte Zahl von Anschlußpunkten von außen keinen direkten Zugang zu jeder einzelnen Einheit, so daß ein auf dem Chip ablaufendes Testverfahren wünschenswert erscheint. Außerdem wird die Testzeit und damit die Testkosten der Chips durch eine dezentrale Diagnose mit parallel auf dem ganzen Netz ablaufenden Tests stark verkleinert. Ein solches dezentrales Testverfahren wird in Kapitel 8 vorgestellt. Dabei bleibt aber folgende Frage offen:

Unter welchen Umständen wird das Rechnernetz immer korrekt repariert?

Betrachten wir ein Netz, das r defekte Einheiten enthält mit $r \geq k(G)$. Dann können gleichartig defekte Einheiten zu dem Diagnoseergebnis gelangen, daß die intakten Einheiten ihrer Nachbarschaft defekt sind und abgeschaltet oder ersetzt werden müssen. Abgesehen von den unnötigen Reparaturkosten kann sich

somit eine Situation ergeben, in der eine korrekte Selbstreparatur des Netzes nicht mehr möglich ist. Um bei maximal t defekten Einheiten sicherzustellen, daß bei einer Reparatur auch eine intakte Einheit mitwirken muß, ist es sinnvoll, folgende Reparaturbedingung einzuführen:

- Für die Reparatur (Austausch bzw. Abschalten) einer Einheit müssen mehr als t Einheiten übereinstimmen.

Dies ist beispielsweise mit einem kryptographischen Code möglich, bei dem t+1 Schlüssel notwendig sind, um das richtige Codewort für das Abschalten oder den Austausch einer Einheit zu errechnen.
Damit wird es für eine Reparatur durch intakte Einheiten notwendig, daß mehr als t intakte Einheiten existieren und ein Nachrichtenaustausch zwischen ihnen möglich ist.
Für eine Klasse von regulären Flächennetzen, deren Randeinheiten zusätzlich mit anderen Randeinheiten verbunden sind, wird in Kapitel 9.3 gezeigt, daß es bei r Defekten, die beliebig in einem Rechnernetz mit N Einheiten verteilt sein können, immer mehr als t intakte, zusammenhängende Einheiten gibt, wenn die Bedingung

$$\sqrt{N} \geq r/2+1$$

erfüllt ist. In diesem Fall können die intakten Einheiten zuerst alle ihre defekten Nachbarn reparieren, dann die Nachbarn der Nachbarn, und so weiter, bis am Schluß das gesamte Rechnernetz fehlerfrei ist.

2.0 Graphentheoretische Ansätze

Sei ein System s gegeben, bestehend aus N Einheiten $u_1..u_N$ mit statistisch unabhängigen Zuverlässigkeiten $R_1..R_N$. Die Einheiten sollen die Eigenschaft haben, sich gegenseitig auf Funktionstüchtigkeit testen zu können mit dem Ergebnis

$$t_{ij}=\begin{cases} 0 & u_i \text{ entdeckte keinen Fehler bei } u_j \\ 1 & u_i \text{ fand einen Fehler bei } u_j \end{cases}$$

Dieses System läßt sich durch einen gerichteten Graphen G(V,E) mit der Knotenmenge V und der Kantenmenge E beschreiben. Die Knoten werden dabei den Einheiten u_i und die Kanten den Testverbindungen im System zugeordnet.
Dieser gerichtete Graph wird 'Testgraph' genannt; die Testergebnisse sind somit Gewichtungen der Kanten. Die geordnete Menge S der |E| Testergebnisse wird als 'Syndrom' bezeichnet, die der defekten Einheiten als 'Fehlerklasse'F.

DEFINITION:

Eine Fehlerklasse F_k heißt 'konsistent' zu einem Syndrom S, wenn beim Auftreten von F_k das Syndrom S beobachtet werden kann.

2.1 Das Grundmodell

Wie in Kapitel 1 geschildert, gaben Preparata et alii in /PRE/ als erste Bedingungen an, unter denen ein System '1-Schritt t-Fehler diagnostizierbar' ('t-diagnostizierbar ohne Reparatur') ist und konstruierten Testgraphen, in denen mit möglichst wenigen Tests möglichst viele Fehler lokalisiert werden können.
Wenn von maximal t defekten Einheiten im Falle eines Defekts mindestens eine identifiziert werden kann, so bezeichneten sie das System mit 'sequentiell t-Fehler diagnostizierbar' ('t-diagnostizierbar mit Reparatur').

Mit den folgenden Annahmen über die Testergebnisse

(2.1a) Wenn die testende Einheit intakt ist, so ist das Testergebnis korrekt (Vollständigkeitsaxiom der Tests).

(2.1b) Wenn die testende Einheit defekt ist, so ist das Testergebnis unabhängig vom Zustand der getesteten Einheit und kann beide Werte, 0 oder 1, annehmen.

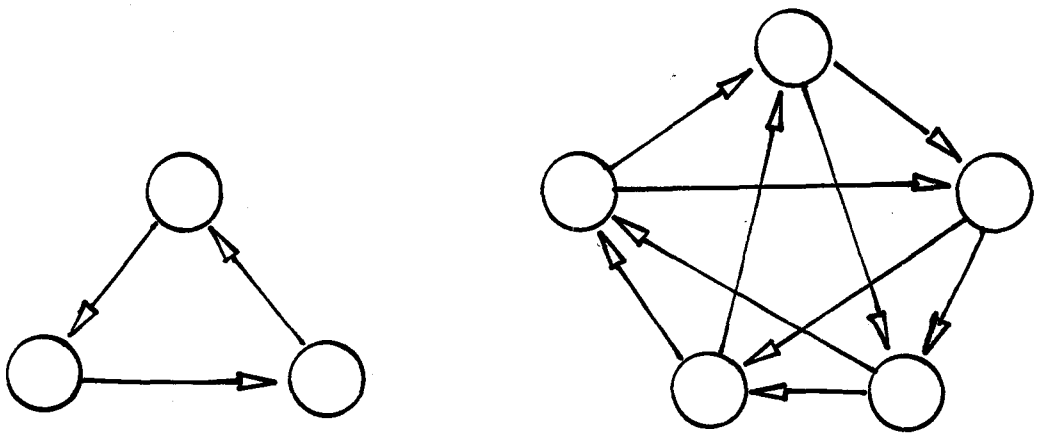
zeigten sie als notwendige Bedingungen

(2.1c) Für 1-Schritt t-Fehler diagnostistizierbare Systeme gilt $N \geq 2t+1$.

Ist umgekehrt $N \geq 2t+1$, so läßt sich immer ein Testgraph angeben, der 1-Schritt t-Fehler diagnostizierbar ist.

(2.1d) Jede Einheit muß mindestens von t andere Einheiten getestet werden.

Mit 2.1c und 2.1d läßt sich eine Klasse D_{1t} von 'optimalen Testgraphen' definieren: Für diese ist $N=2t+1$ und jede Einheit wird von genau t anderen Einheiten gestestet. Abb. 2.1a zeigt Beispiele dazu.



D_{11} D_{12}

Abb. 2.1a Beispiele für optimale Testgraphen

Bei 1-Schritt t-Fehler diagnostizierbaren Systemen ist die Zahl der Testverbindungen relativ hoch; bei D_{1t}-Graphen sind es $m=Nt=N+N(t-1)$. Für die weniger Information benötigenden sequentiell t-Fehler diagnostizierbaren Systeme zeigten sie, daß es eine Klasse von Testgraphen mit der weit geringeren Zahl von $m= N+2(t-1)$ gerichteten Testverbindungen gibt.

Sei die Menge der von der Einheit u_i getesteten Einheiten mit $D(u_i)$ notiert, wobei für alle $u_j \in D(u_i)$ gilt $(u_i,u_j) \in E$.

Bei einer Menge X soll $D(X):=\{u_j/(u_i,u_j)\in E, u_i \in X, u_j \notin X\}$ bedeuten.

Zu den notwendigen Bedingungen für 1-Schritt t-Fehler Diagnose fanden Hakimi und Amin /HAK/ auch die hinreichenden Bedingungen:

Der Testgraph G ist d.u.n.d. 1-Schritt t-Fehler diagnostizierbar, wenn gilt:

(2.1c), (2.1d) und

(2.1e) Für alle $p \in \{0,1,\ldots,t-1\}$ und jede Teilmenge $X \in V$ mit $|X|= N-2t+p$ Knoten muß für die Menge D(X) der von X Getesteten die Relation $|D(X)| > p$ erfüllt sein.

Eine graphentheoretische Formulierung mittels Partitionen ist in /ALL/ zu finden.

Alternativ dazu zeigten Hakimi und Amin:

(2.1f)

Seien 2.1a und 2.1b gegeben.

Sei G ein Testgraph, in dem keine zwei Einheiten existieren, die sich gegenseitig testen.

G ist genau dann t-diagnostizierbar ohne Reparatur, wenn in G die Bedingung 2.1d gilt.

Außerdem gaben sie eine weitere, hinreichende Bedingung für t-Diagnostizierbarkeit ohne Reparatur an, die den Zusammenhang K(G) eines Graphen, d.h. die kleinste Zahl von Knoten, deren Entfernung den Graphen nicht mehr streng zusammenhängen läßt, benutzt:

(2.1f') Sei $N \geq 2t+1$.

Wenn $K(G) \geq t$ ist, so ist G t-diagnostizierbar ohne Reparatur.

In einem Anwendungsbeispiel zeigten Armstrong und Gray /ARM/, daß eine Verbindung von Prozessoren in Form eines n-dim. Bool'schen Würfels erst dann einen isolierten Knoten (Prozessor) hat, wenn a Knoten und b Kanten ausfallen, wobei $a+b \geq n$ ist. Also ist K(Bool'scher Würfel)=n. Da auch für $n \geq 3$

$2^n \geq 2n+1$ gilt, ist der Bool'sche Würfel der Dimension $n \geq 3$ n-diagnostizierbar ohne Reparatur.

U.Manber zeigte in /MAN/ weitere Bedingungen für t-diagnostizierbare, streng zusammenhängende (s.u.) Graphen mit Reparatur durch Einführung von sog. 'Detektionsmengen D_i' von Einheiten, die sich dadurch auszeichnen, daß in diesen Mengen exklusiv entweder eine bestimmte Einheit oder aber alle anderen defekt sind. Mit diesem Einteilungskonzept konnte er zeigen, daß für alle t-diagnostizierbaren Graphen G(V,E) mit Reparatur gilt, daß

G(V,E) ist t-diagnostizierbar mit Reparatur
... also ist $|V| > 2t$.
... d.u.n.d., wenn für jedes Syndrom mit mindestens einer konsistenten Fehlerklasse s Detektionsmengen $D_i, i=1..s$ existieren mit

$$\bigcap_i D_i = \emptyset$$

so daß $(s-1)+(|D_{max}|-1) > t$ ist. Dabei ist $|D_{max}| := \max_i |D_i|$

Mit den hinreichenden Bedingungen

a) $|V| > 2t$
b) ex eine Einheit, die von t anderen Einheiten getestet wird
c) jede dieser t Einheiten wird von mindestens einer anderen, verschiedenen getestet.

definiert er eine Klasse von Graphen, die t-diagnostizierbar mit Reparatur sind. Ein Beispiel zeigt Abb.2.1b für t=4, $|V| = 9$.

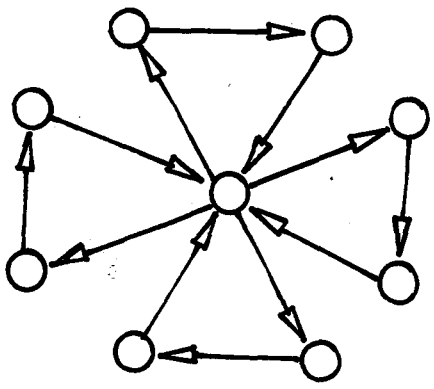


Abb.2.1b 4-diagnostizierbarer Testgraph mit Reparatur

Weitere Überlegungen sind in /CIO/ und /SAH/ zu finden.

Von R.F. Madden wurde in /MAD/ ein Diagnosealgorithmus vorgestellt, der in polynomialer Laufzeit mit der Komplexität $O(N^3)$ eine Diagnose ausführt.

2.2 Ein modifiziertes Grundmodell

Das grundlegende Modell von Preparata, Metze und Chien wurde von verschiedenen Autoren durch Zusatzannahmen verändert.
Eine der ersten wurde von Barsi, Grandoni und Maestrini in /BAR1/ vorgestellt:
(2.2a)

> Wenn beide Einheiten defekt sind, so lautet das Testergebnis immer 'defekt'.

Mit dieser Annahme ergeben sich als notwendige und hinreichende Bedingungen für t-Diagnostizierbarkeit ohne Reparatur wieder 2.1d und eine weniger restriktive Version von 2.1f:
Seien die eine Einheit u_i testenden Einheiten mit $B(u_i)$ notiert, d.h. für alle $u_j \in B(u_i)$ gilt $(u_j,u_i) \in E$.

(2.2b) Wenn zwei Einheiten u_i und u_j existieren, die sich gegenseitig testen und jeweils von genau t getestet werden, so muß eine dritte Einheit u existieren, für die entweder $u \in B(u_i)$, $u \notin B(u_j)$, $B(u) \neq B(u_j)$ oder $u \in B(u_j)$, $u \notin B(u_i)$, $B(u) \neq B(u_i)$ gilt.

Mit der letzten Bedingung vermeidet man die auch mit (2.2a) ohne weitere Testergebnisse nicht diagnostizierbare Situation von zwei Einheiten, deren Testergebnis für die andere Einheit jeweils 'defekt' ist.
Analog zu Preparata et alii definierten sie eine Klasse von t-diagnostizierbaren Testgraphen als 'optimal', wenn jede Einheit gerade von t anderen Einheiten getestet wird. Infolgedessen enthält die Klasse der optimalen Testgraphen im Sinn von Barsi et alii alle Testgraphen mit dem Verbindungschema der D_{1t}-Graphen und beliebigem N und t, also die D_{1t}-Graphen von Preparata mit $N=2t+1$ und zusätzlich die Testgraphen mit geradem N und $N-2 > t \geq N/2$.
Auch für t-Diagnostizierbarkeit mit Reparatur finden sie eine neue, hinreichende Bedingung:
(2.2c)

> Der Testgraph ist streng zusammenhängend, d.h. zwischen je zwei Einheiten existiert eine gerichtete Testverbindung.

(2.2d) In jeder Teilmenge von V mit N-t Einheiten existiert eine Einheit, die von einer anderen Einheit der Teilmenge getestet wird.

Die Klasse der optimalen, t-diagnostizierbaren Testgraphen mit Reparatur definierten sie mit

> Ein t-diagnostizierbarer Testgraph mit Reparatur ist optimal, wenn $m = t + \lceil (N-1)/2 \rceil + 1$ Testverbindungen bestehen.

Eine Teilmenge dieser optimalen Graphen bilden die sog. k-Rosetten, also Graphen, bei denen k Kreisketten zusammenhängen. Bei einer durch eine weitere Bedingung eingeschränkten Teilmenge ist $t = k + \lfloor N/2 \rfloor - 2$.
Ein Beispiel dafür ist in Abb.2.2a zu sehen.

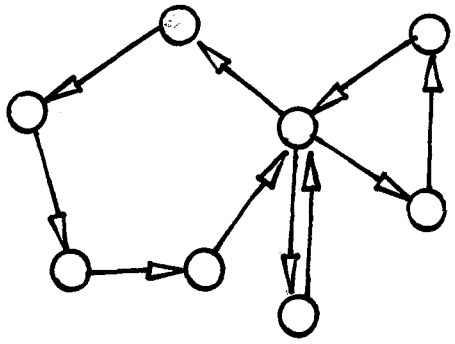


Abb.2.2a Ein optimales System mit t=5

Ein Algorithmus der Komplexität $O(N^3)$, der maximales t eines Testgraphen bestimmt, sowie ein Diagnosealgorithmus der Komplexität $O(N^2)$ ist in /AMM2/ zu finden.

2.3 Das Vergleichstestmodell

Gegenüber dem Grundmodell, das über Testergebnisse nur bei einer intakten testenden Einheit Aussagen macht (2.1a), nahm Malek in /MAL/ statt (2.1b) an, daß

(2.3a) das Testergebnis immer $t_{ij}=1$('defekt') lautet, wenn die testende Einheit defekt ist.

Wenn zwei Testergebnisse von Tests zwischen den Einheiten u_i und u_j vorliegen folgt aus 2.3a

$$t_{ij}=t_{ji}$$

Mit den Modellannahmen läßt sich also nach Ausführung eines Tests t_{ij} sofort auch das Testergebnis t_{ji} voraussagen, ohne daß der Test durchgeführt werden muß.Es braucht bei diesem Modell keine Testrichtung angegeben werden; der Testgraph ist ungerichtet ($d_{out}=d_{in}=:d$). Diese Symmetrie läßt sich als 'Vergleich' der Zustände der beiden Einheiten deuten, der verwendete Test ist ein 'Vergleichstest'. Damit kann man das Modell auch zur Fehlerdiagnose bei funktionsbegleitenden Tests verwenden; beispielsweise lassen sich die Einheiten u_i als gleichartige, redundante Programme auffassen, deren auf verschiedenen Computern errechneten Resultate durch Vergleich on-line auf ihre Richtigkeit überprüft werden.

Für Fehlerkennung gab Malek nur Ober- und Untergrenzen an. Genauere Beziehungen wurden von Ammann und Dal Cin in /AMM/ vorgestellt. Sie fanden die zu (2.2b) parallele Beziehung:

Seien 2.1a und 2.3a vorausgesetzt.
Sei $N(u_i)$ die Menge aller Nachbarknoten, die sich mit u_i vergleichen.
Ein Testgraph $G=(K,E)$ ist genau dann t-diagnostizierbar (ohne Reparatur), wenn

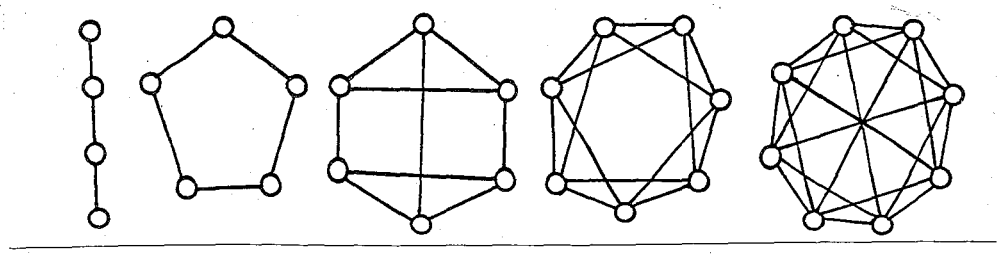
1) $d \geq t$
2) Für jeden Test t_{ij} mit $|N(u_i)|=|N(u_j)|=t$ existiert eine Einheit u mit entweder $u \in N(u_i)-N(u_j)$, $N(u)\neq N(u_j)$ oder $u \in N(u_j)-N(u_i)$, $N(u)\neq N(u_i)$.

Dies bedeutet :

a) G ist t-diagnostizierbar => $t\leq n-2$, $|E|\geq\lceil nt/2\rceil$
b) G ist t-diagnostizierbar, $|E|=\lceil nt/2\rceil$ => G ist t-optimal(s.u.).
c) $d \geq t+1$ => G ist t-diagnostizierbar.

Sie zeigten weiterhin, daß optimale Testgraphen bei gegebenem t und mit

möglichst geringer Kanten- und Knotenzahl dadurch konstruiert werden, daß vom vollständig vernetztem Graphen mit N=t+2 Knoten eine Kante gestrichen wird. Zusätzlich gaben sie Konstruktionsvorschriften für streng t-optimale Testgraphen an, d.h. Testgraphen, in denen bei nicht mehr als t defekten Einheiten diese dadurch gefunden werden können, daß die Intakten durch t_{ij} ='intakt' identifiziert werden können und bei gegebener Knotenzahl möglichst geringe Kantenzahl besitzen.
Nachstehende Abbildungen zeigen einige Beispiele für streng t-optimale Graphen.



Ein Diagnosealgorithmus der Komplexität $O(N^2)$ ist in /AMM2/ zu finden.
Ein anderes Vergleichstestmodell ist in /MAE/ enthalten.

2.4 Diagnose mit präventiver Reparatur

Bei gegebenem Testgraphen läßt sich im Modell von Preparata et alii (s.2.1) in einem Schritt keine eindeutige Diagnose mehr durchführen, wenn mehr als t Einheiten defekt sind. Wenn man aber zuläßt, auch nicht defekte Einheiten zu ersetzen, so lassen sich einerseits bei der 1-Schritt Diagnose Testverbindungen oder andererseits bei sequentieller Diagnose Testzeit einsparen. Man kann nun ein neues $t' > t$ definieren, bei dem die Fehler im System zwar nicht mehr in einem Schritt korrekt diagnostiziert, aber doch repariert werden können. Mit diesem Gedanken definierte Friedman /FRI/ :

> Ein System ist k-Schritt t/s diagnostizierbar, wenn in k Anwendungen einer Diagnoseprozedur jede Fehlerklasse mit $\leq t$ Fehlern diagnostiziert und das System durch Ersetzen von maximal s Einheiten repariert werden kann.

In /KAR1/,/KAR2/ zeigte er mit Karunanithi, daß für ein Ringsystem gilt:

> Ein Ringsystem ist 1-Schritt t/s diagnostizierbar

- d.u.n.d., wenn $N \geq s+2$ ('optimales System': $N=s+2$) gilt mit $s=\lceil t/2 \rceil \lfloor t/2 \rfloor + t$
- d.u.n.d., wenn es sequentiell t-Fehler diagnostizierbar ist.

Die Ursache der zweiten Beziehung liegt darin, daß in beiden Diagnosen die Existenz und Lokalisierung von mindestens einer defekten Einheit vorausgesetzt wird. Die sequentielle Diagnose ersetzt diese fehlerhafte Einheit und testet erneut, während die t/s-Diagnoseprozedur die ungünstigste Fehlerklasse, die die die fehlerhafte Einheit enthält, berechnet und und in einem Schritt alle betroffenen Einheiten ersetzt.
Durch Weglassen von Testverbindungen im D_{1t} -Graphen läßt sich eine neue Klasse D_{1A} von Testgraphen definieren, die mit $A \leq t$ die optimalen t-diagnostizierbaren Testgraphen von Preparata als Spezialfall bei A=t enthält.

In Abb.2.4a ist einem D_{1t} -Graphen mit t=3,N=7 ein D_{1A}-Graph mit A=2,N=7 gegenübergestellt.

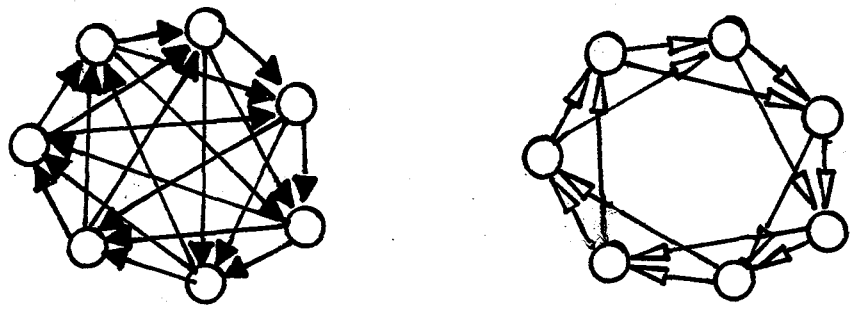


Abb. 2.4a D_{13} D_{12}, N=7

Für diese D_{1A}-Graphen konnte Friedman zeigen, daß sie bei $N \geq 2t+1$ und $A > \lfloor t/2 \rfloor$

---- sequentiell t-Fehler diagnostistizierbar in maximal 2 Schritten und

---- 1-Schritt t/s diagnostizierbar mit s=2t-A

sind.

2.5 Eine einfache Diagnose

Die oben erwähnten Gedanken von Friedman über t/s-Diagnostizierbarkeit benutzte Smith /SMI/ zu dem folgenden, verblüffend einfachen Diagnosealgorithmus:
(2.5a)

1) Führe alle Tests t_{ij} durch. Notiere die Anzahl der Tests, die '1' für die Einheit u_j ergeben, am j-ten Platz

$$\sigma_j := \sum_i t_{ij}$$

des Aggregatsyndroms $\sigma=(\ldots,\sigma_j,\sigma_{j+1},\ldots)$.
Falls $\sigma=(0\ldots0)$, STOP.

2) Wähle aus der Menge $F=\{u_j / \Sigma t_{ij}\neq 0$ und u_j wurde nicht ersetzt $\}$ eine Teilmenge F' aus und ersetze alle Einheiten von F'.
Gehe nach 1)

Für die Teilmenge F' bieten sich zwei verschiedene Strategien an:

STRATEGIE 1: Wähle F':=F, d.h. ersetze alle zweifelhaften Einheiten.
STRATEGIE 2: Wähle $F':=\{u_j | \Sigma t_{ij}=max\}$ die Einheiten u_j, die nach der Mehrheitsentscheidung defekt sind.

Er zeigte, daß in jedem Fall der Algorithmus terminiert.
Für die Effizienzuntersuchung des obigen Algorithmus führte er das Maß 'f/s-diagnostizierbar' ein:

Ein System ist f/s-diagnostizierbar, wenn bei f fehlerhaften Einheiten maximal s ersetzt werden, um alle Defekte zu reparieren.

Wie groß ist s für ein gegebenes System?
Dazu werden zwei Fälle unterschieden :

1) 'symmetrische Invalidierung'(Preparata):
Das Testergebnis einer defekten Einheit ist unabhängig von der getesteten Einheit.

2) 'asymm. Invalidierung'(Barsi):
Wenn die testende Einheit defekt und die getestete Einheit intakt ist, so ist das Testergebnis unbekannt; ist die getestete Einheit dagegen defekt, so lautet das Testergebnis 'defekt'.

Seien T_{max} die maximale Zahl der Tests, die jede Einheit durchführt und T_{min} die minimale Zahl der Tests, mit der jede Einheit getestet wird.
Dann gilt für Strategie 1:
$s = (T_{max}+1)f$ bei symm. und asymm. Invalidierung (z.B. Kreiskette: s=2f)

und für Strategie 2 bei symm.Invalidierung und $T_{min} \leq T_{max}$:
$s = (T_{max}-T_{min}+2)f$ ist optimal bei $T_{max}=T_{min}$, $s=2f$
und bei asymm.Invalidierung: $s=(|T_{max}/T_{min}|+1)f$

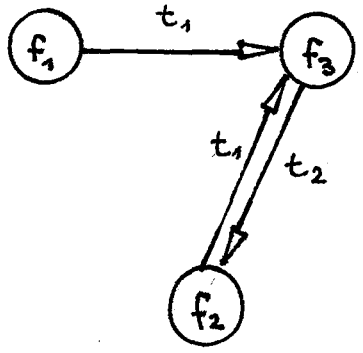
als obere Schranke.
Genauer wurde die Effizienz von Butler /BUT/ für die D_{1A}-Graphen von Friedman untersucht. Er fand einen typischen Unterschied zwischen beiden Strategien: In Stragie 1 ist die mittlere Zahl von falsch ersetzten Einheiten hoch, die Zahl der Testzyklen dagegen niedrig, während dies bei Strategie 2 umgekehrt ist. Je nach Testkosten und Reparaturkosten muß also für ein diskretes System abgewogen werden, welcher von beiden Strategien der Vorzug gegeben werden soll.
Weitere Diagnosealgorithmen sind auch in /KAM/ und /ADH/ zu finden.

2.6 Diagnose bei ungleichen Einheiten

Im Modell von Preparata et alii /PRE/ wurde von den Einheiten angenommen, daß sie selbstständig andere Einheiten, z.B. Computersysteme, testen können. Das Modell von Russel und Kime /KIM2/ /KIM3/ verallgemeinert diese Annahme dergestalt, daß mehrere Einheiten für den Test einer Einheit notwendig sind. Fällt eine der benötigten Einheiten aus, so ist das Testergebnis nicht mehr korrekt.
Diese Annahme spiegelt die Tatsache wieder, daß z.B. auf Computerebene bei dem Test eines RAM die CPU, der Systembus und ein ROM, also mehrere Einheiten, intakt sein müssen, um ein korrektes Testergebnis zu erhalten. Die von Preparata betrachteten Systeme sind damit als Spezialfall im Ansatz enthalten.
Dazu gehen Russel und Kime folgendermaßen vor:

$$C = \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} \quad T = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Über der Grundmenge aller n Fehler (bzw evtl.fehlerhafter Einheiten) f_i und aller p Tests t_j werden verschiedene Matrizen definiert. In der C-Matrix der Dimension n×p sind für jeden Fehler die Tests angegeben, die die Existenz des Fehlers unter der Voraussetzung zeigen, daß es der Einzige im System ist. Im nebenstehendem Beispiel ist der Test t_2 für f_2 nur gültig, wenn f_3 nicht defekt ist. Außerdem wird eine Matrix T der Dimension n×p definiert, in der für jeden Test verzeichnet wird, welche fehlerhaften Einheiten ihn ungültig machen, d.h. welche Einheiten an seiner Ausführung beteiligt sind.



Sind beide Matrizen gegeben, so läßt sich leicht daraus der entsprechende Testgraph konstruieren, da ja für die Adjazenzmatrix $A=T*C^{transp}$ gilt. Es gibt aber Systeme (z.B.TMR), bei denen diese I-formationen nicht ausreichen, da Tests existieren, die nicht durch einzelne Fehler ungültig werden ('morphische Systeme'), sondern nur durch bestimmte, gleichzeitig auftretende Fehlerkombinationen ('semimorphische Systeme'). Deshalb wird zusätzlich eine generelle Fehlertafel mit der Matrix G der Dimension 2^n×p definiert, deren Eintragungen sich nicht im Testgraphen widerspiegeln:

$$G_{kj} := \begin{cases} 0 \\ 1, \text{ wenn im Fehlerzustand } F_k \text{ das Testergebnis } t_j \begin{matrix} \text{'intakt'} \\ \text{'Fehler' ist.} \\ \text{eines davon} \end{matrix} \\ X \end{cases}$$

Mit der G-Matrix lassen sich nun hinreichende und notwendige Bedingungen für t-Diagnostizierbarkeit mit und ohne Reparatur formulieren. Um eine größere Einsicht in die strukturellen Zusammenhänge zu gewinnen, definierten Russel und Kime verschiedene Strukturmaße:

c(sys)

Der Geschlossenheitsindex des Systems sys ist die Zahl der Fehler in der kleinsten geschlossenen Fehlerklasse des Systems. Eine Fehlerklasse F_k ist geschlossen, wenn jeder Test für f_i aus F_k durch ein f_j aus F_k ungültig wird, also kein verläßliches Testergebnis für F_k existiert.

Für c(sys) läßt sich zeigen, daß $g(G) \leq c(sys) \leq r(G)$ gilt, wobei im Graphen G g(G) die Länge des kleinsten gerichteten Zyklus und r(G) die Zahl der Knoten, die alle anderen Knoten erreichen können, darstellen.

Für die Fehlerwechselwirkung definieren sie

$m(F_j)$ Der Maskierungsindex einer Fehlerklasse F_j ist die Zahl der Fehler in der kleinsten Fehlerklasse, die alle Tests für F_j ungültig macht.

m(sys) heißt der kleinste Maskierungsindex im System.

$e(F_j)$ Der Expositionsindex der Fehlerklasse F_j ist die Zahl der Fehler in F_j, die durch Anwesenheit der anderen f_i aus F_j nicht maskiert werden, d.h. es gibt mindestens einen Test, der für erstere noch gültig ist.

$e_k(sys)$
ist der kleinste Expositionsindex im System, wobei nur Fehlerklassen mit k Fehlern betrachtet werden.

Mit diesen drei Grundindizes lassen sich unter anderem folgende Sätze herleiten:

-- Wenn ein morphisches System t-diagnostizierbar mit Reparatur ist, so ist $c(sys) \geq 2t+1$.

-- Ein morphisches System ist genau dann t-diagnostizierbar ohne Reparatur, wenn

1) $c(sys) \geq 2t+1$
2) $m(sys) \geq t$
3) $e_k(sys) \geq 2t+1-k$ für $k=t+1,..,\min(2t-1,n)$

Für semimorphische Systeme gelten ähnliche Sätze.
Das Modell von Russel und Kime ist durch seine Fülle von notwendig zu spezifizierender Information sehr groß und unhandlich für größere Systeme; schon bei 16 Einheiten, für die mindestens 16 Tests benötigt werden, ist die G-Matrix schon von der Größe 2^{16} *16 Bit, also von der Größe des maximal direkt adressierbaren Speichers eines 16-Bit Rechners. Deshalb ist dieses Modell auch nach Meinung seiner Autoren nur als Entwicklungsmodell (vgl. auch /KIM1/,/KIM4/), nicht aber zur realen Diagnose geeignet und soll deshalb im Folgenden nicht weiter betrachtet werden.

3.0 Wahrscheinlichkeitstheoretische Ansätze

Im vorigen Abschnitt wurden einige Arbeiten präsentiert, die nur mit wenigen, grundsätzlichen Annahmen über Einheiten und Testergebnisse (2.1a,b) zu Diagnosealgorithmen und Aussagen über optimale Testgraphen kamen. Dabei wurde durchweg als Kriterium für t-Diagnostizierbarkeit ohne Reparatur gewählt, daß es kein Syndrom geben darf, das von mehreren möglichen Fehlerklassen der Kardinalität $\leq t$ erzeugt sein kann.
Obige Voraussetzung läßt sich aufgeben, ohne daß es dabei zu unentscheidbaren Diagnosesituationen kommt, wenn man zusätzliche Informationen über die Zuverlässigkeiten der Einheiten ausnutzt.

3.1 Diagnose bei Kenntnis der Zuverlässigkeit

Maheshwari und Hakimi gingen in /MAH/ dazu folgenden Weg:
Angenommen, die Auftrittswahrscheinlichkeiten der Fehlerklassen seien bekannt und die Voraussetzungen 2.1a und b seien gegeben. Dann ist die am wahrscheinlichsten vorliegende Fehlerklasse F diejenige, die mit dem beobachteten Syndrom konsistent ist und die größte Auftrittswahrscheinlichkeit P(F) hat:

$$P(F) := \max_{F_i} P(F_i) \qquad F_i \text{ ist konsistent zum Syndrom}$$

Im allgemeinen kann es mehrere F_i geben, die obige Bedingung erfüllen und gleiches $P(F_i)$ haben.
Damit ist das Syndrom mindestens mit der Wahrscheinlichkeit P(F) richtig diagnostiziert. Wenn es nun eine Zahl T gibt, so daß für alle Syndrome einer Fehlerklasse F die Beziehung $P(F) \geq T$ gilt, so werden diese Syndrome ebenso mindestens mit der Wahrscheinlichkeit T richtig diagnostiziert.
DEFINITION:

> Ein System heißt probabilistisch T-diagnostizierbar (p-T diagnostizierbar), wenn für jedes Syndrom im gerichteten G=(V,E) maximal EINE konsistente Fehlerklasse F mit P(F)>T existiert.

Wie hängt nun die Struktur des Testgraphen mit der p-T Diagnostizierbarkeit zusammen? Dazu müssen wir statt der Fehlerklasse F die einzelnen Einheiten betrachten.
Angenommen, die Einheiten u_i haben statistisch unabhängige Zuverlässigkeiten

R_i, die bekannt sind. Sei e_i definiert mit

$$e_i := \begin{cases} 0 & u_i \text{ intakt} \\ 1 & u_i \text{ defekt} \end{cases}$$

Dann ist

$$P(F) = \prod R_i^{1-e_i} \prod (1-R_i)^{e_i}$$

die Produktwahrscheinlichkeit aus den Einzelwahrscheinlichkeiten. Bildet man nun den Logarithmus $\ln(P(F))$, so ergibt sich statt des Produkts eine Summe, und nach einigen Umformungen ist

$$P(F) > T \quad <=> \quad \sum_{u_i \varepsilon F} \ln\left(\frac{R_i}{1-R_i}\right) \quad < K(T)$$

mit

$$K(T) := -\ln(T) + \sum_{u_i \varepsilon V} \ln(R_i)$$

als Konstante. Führt man das Gewicht $W(u_i) := \ln(R_i/1-R_i)$ für jede Einheit u_i ein, das bei $R_i > 1/2$ positiv ist, so ist nur für die eine konsistente Fehlerklasse F die Beziehung

$$W(F) := \sum_{u_i \varepsilon F} W(u_i) < K(T)$$

erfüllt.

Die Diagnose für p-T diagnostizierbare Graphen beschränkt sich also darauf, bei gegebenem Syndrom eine dazu konsistente Menge F von Einheiten so zu finden, daß für deren Gewichtssumme obige Relation gilt und damit eine richtige Diagnose mit ausreichender Wahrscheinlichkeit sichert.

Als dafür notwendige Bedingung in p-T-diagnostizierbaren Testgraphen finden Maheshwari und Hakimi, daß es im System keine Einheit geben darf, die geringe Zuverlässigkeit besitzt und von ebenfalls unzuverlässigen Einheiten getestet wird: Für jedes u_i muß $W(u_i) + W(B(u_i)) \geq K(T)$ sein.

In Systemen, in denen zwei Einheiten sich nicht gegenseitig testen können (vgl. 2.1f) ist obige Bedingung in der abgewandelten Form

$$1/2\ W(u_i) + W(B(u_i)) \geq K(T)$$

auch hinreichend.

3.2 Konstruktion der dazu optimalen Testgraphen

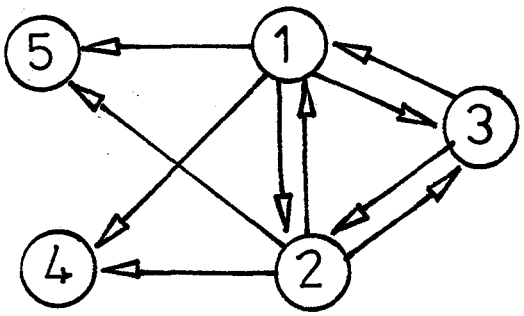
Von Fujiwara und Kinoshita wurden die obigen Gedanken in /FUI1/ folgendermaßen weitergeführt:
Sei eine 'Basismenge' U so definiert, daß keine Teilung von U in U_1 und U_2 existiert mit der Eigenschaft $W(U_1)<K(T)$ und $W(U_2)<K(T)$. Sie zeigten, daß für ein System genau dann ein p-T-diagnostizierbarer Digraph (mit und ohne Reparatur) existiert, wenn die Knotenmenge V die Eigenschaften einer Basismenge hat. Dazu spezifizierten sie zwei Design D_0 bzw. D_1, die mit Hilfe einer gegebenen Basismenge UcV für V Testgraphen konstruieren, die p-T diagnostizierbar sind:

Design D_0:

1) Bilde den kompletten Digraph <U> zu vorliegender Basismenge U
2) Verbinde die restlichen Einheiten $u_i \varepsilon$ V-U so mit den Einheiten der Basismenge, daß

$$B(u_i) \subseteq U \text{ und } W(B(u_i)) \geq K(T) .$$

Falls dies möglich ist, hat der entstandene Graph das Design D_0 und ist p-T diagnostizierbar ohne Reparatur.



$R_1=6/7$ $W_1=\ln(6)=1.79$
$R_2=5/6$ $W_2=\ln(5)=1.609$
$R_3=4/5$ $W_3=\ln(4)=1.386$
$R_4=3/4$ $W_4=\ln(3)=1.098$
$R_5=2/3$ $W_5=\ln(2)=0.693$
$T=1/35$
$K=\ln(35)+\ln(2)-\ln(7)=2.302$

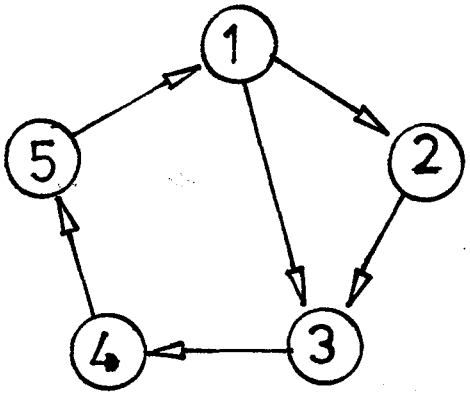
Beispiel zum Design D_0 mit $U=\{u_1,u_2,u_3\}$

Seien die N Einheiten von V so numeriert, daß die Einheiten $u_i, i=1..s$ mit $s:=|U|$ in U liegen.

Design D_1:

1) Bilde einen Ring mit $(u_i,u_j)\varepsilon E$, wobei $j:=i+1 \bmod N$ gilt.
2) Alle Einheiten der Basismenge testen die Einheit u_s:
   Für $1 \leq i < s$ ist $(u_i,u_s)\varepsilon E$

Dieser so entstandene Testgraph ist p-T diagnostizierbar mit Reparatur.



Beispiel zum Design D_1 mit $U=\{u_1,u_2,u_3\}$

In der nachfolgenden Arbeit /FUI2/ beschäftigten sie sich zusätzlich mit den notwendigen und hinreichenden Bedingungen für p-T Diagnostizierbarkeit, wenn zusätzlich Bedingung (2.2a) vorausgesetzt wird und formulierten Sätze analog zu /BAR/.

3.3 Diagnose bei Kenntnis aller Parameter

In seiner Arbeit /BLO/ setzte Blount wie bisher nicht nur die Kenntnis der Lebensdauer der Einheiten voraus, sondern auch die Wahrscheinlichkeiten, mit der diese Syndrome erzeugen. Dies ist im Vergleich mit den bisher benutzten Annahmen in Tabelle 1 aufgeführt.

Zustand der Einheiten u_i	u_j	Test-ergebnis t_{ij}	$P(t_{ij}/(e_i e_j))$ pro Modell Blount	/Preparata	/Barsi	/Malek
0	0	0	p_{ij}	1	1	1
		1	$1-p_{ij}$	0	0	0
0	1	0	$1-r_{ij}$	0	0	0
		1	r_{ij}	1	1	1
1	0	0	q_{ij}	p	p	0
		1	$1-q_{ij}$	1-p	1-p	1
1	1	0	$1-s_{ij}$	p	0	0
		1	s_{ij}	1-p	1	1

Tabelle 1

Dabei ist $P(t_{ij}/(e_i\ e_j))$ die Wahrscheinlichkeit für das Auftreten des Testergebnisses $t_{ij} \varepsilon \{0,1\}$, wenn der Zustand $(e_i e_j)$ $\varepsilon\{(00),(01),(10),(11)\}$ der Einheiten u_i und u_j vorliegt. Entsprechende Parameter führte er auch für Selbsttests ein.
Diese zusätzlichen Annahmen sollen die Tatsache widerspiegeln, daß in der Praxis meist keine vollständigen Tests ($r_{ij} \neq 1$) existieren und die Tests durch Übersprechen von Signalleitungen etc. gestört sein können ($p_{ij} \neq 1$).
Mit der Definition der Diagnostizierbarkeit D_{sys} als Wahrscheinlichkeit dafür, daß ein vorliegender Defekt bei gegebener Diagnosevorschrift σ auch richtig diagnostiziert wird, ist

$$D_{sys} := \sum_k P(F_k \text{ erkannt}/F_k)\ P(F_k)$$

Sei nun jedem der 2^M Syndrome S_l eine Fehlerklasse mittels einer Diagnosevorschrift σ zugeordnet worden. Dann ist

$$P(F_k \text{ erkannt}/F_k) = \sum_{S_l \text{ mit } \sigma(S_l)=F_k} P(S_l/F_k)$$

und damit ist

$$D_{sys} = \sum_k \sum_{S_l \text{ mit } \sigma(S_l)=F_k} P(S_l/F_k)P(F_k)$$

Eine optimale Diagnosevorschrift maximiert D_{sys}. Dies ist erreicht, wenn σ mit

$$P(S_i,\sigma(S_i)) \stackrel{!}{=} \max_k P(S_i/F_k)\ P(F_k)$$

definiert wird. Die obige Formel in algorithmischer Form stellt bereits den Diagnosealgorithmus dar.
Da Blount vollkommen auf graphentheoretische Modellbezüge verzichtet, fallen damit graphentheoretische Beschränkungen, aber auch die Möglichkeit weg, mittels verbesserter Testgraphen auch D_{sys} zu verbessern. Die zu Grunde liegenden Testgraphen gehen nur indirekt über $P(S_i/F_k) \neq 0$ in die Diagnose ein. Die Vorteile dieses Diagnoseansatzes werden in Teil 5 näher untersucht, die Nachteile liegen in dem Aufwand, unter allen 2^M Fehlerklassen diejenige mit maximalem $P(S_i,F_k)$ zu finden. Näheres über die Komplexität dieser Diagnose ist in Anhang A zu finden.

3.4 Iterative Diagnose

Bossen und Hsiao beschäftigten sich in /BOS/ näher mit der iterativen oder sequentiellen, probabilistischen Diagnose mit Reparatur und gaben verschiedene Diagnosestrategien dafür an, die im Folgenden kurz wiedergegeben werden sollen. Seien N unabhängige Einheiten u_i eines Systems gegeben. Angenommen, es wird ein Fehler entdeckt und er kann mit Hilfe des beobachteten Syndroms S_j nicht ausreichend lokalisiert werden. Dann soll eine Teilmenge der Einheiten als defekt diagnostiziert und ersetzt werden.
Bemerkenswert an diesem Verfahren ist die Tatsache, daß keine gleichen oder aktiv testenden Einheiten vorausgesetzt werden; zur Diagnose reicht die Angabe der bedingten Wahrscheinlichkeiten der Syndrome aus.
Für dieses Diagnoseverfahren werden mehrere Strategien angegeben:

1) Deterministische Diagnose:
   Alle defekten Einheiten können eindeutig in einem Schritt identifiziert werden.

Dies ist sicher nicht immer möglich.

2) 2-Schritt Strategie A:

1.Schritt: Ersetze die Einheit(en) mit
$P(S_j/e_i=1)\ P(e_i=1) = \max_i$
2.Schritt: Falls der Fehler weiterhin existiert,
ersetze alle anderen Einheiten.

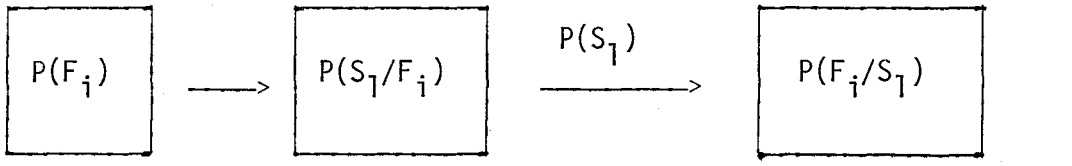
3) 2-Schritt Strategie B:
1.Schritt: Ersetze die Einheiten mit größtem
$P(S_j/e_i=1)P(e_i=1)$ und $u_i \varepsilon F$, so daß
$P(S_j/F)$ größer als eine Schwelle,
z.B. 90 Prozent, ist.
2.Schritt: Falls weiterhin Fehlermeldung, ersetze alle anderen
Einheiten.

4) sequentielle probabilistische Diagnose:
1.Schritt: Ersetze eine Einheit wie in Strategie A, 1.Schritt.
2.Schritt: Falls weiterhin eine Fehlermeldung erfolgt,
ersetze von den restlichen Einheiten wieder
diejenige mit maximalem $P(S_j,e_i=1)$.
...
so lange, bis keine Fehlermeldung oder keine unersetzten
Einheiten mehr existieren.

Man sieht, daß die verschiedenen Strategien sich durch verschiedene Kosten unterscheiden. In Strategie 2) und 3) werden unnötig Einheiten ersetzt, dagegen in 4) mehr Zeit benötigt.

4.0 Ansätze der Statistik und Mustererkennung

In dem vorigen Abschnitt wurden, ausgehend von einigen Grundgedanken, einige Ansätze für Wahrscheinlichkeitsdiagnosen vorgestellt. Im Folgenden soll nun die Diagnosesituation durch Vorstellung anderer grundsätzlicher Ansätze gründlicher ausgeleuchtet werden.
Wie in Kapitel 1 dargelegt worden ist, liegt bei der Diagnose von Syndromen S_j, die mit der bedingten Wahrscheinlichkeit $P(S_l/F_i)$ von dem Fehlerzustand F_i erzeugt werden, folgende Situation vor:



Eine Fehlerquelle produziert ein Syndrom das beobachtet und diagnostiziert wird.

Um diese Situation näher zu untersuchen, sei in der Terminologie der Mustererkennung statt der Fehlerklasse F_i die Musterklassen mit ω_i und statt des Syndroms S_l die beobachteten Muster mit x bezeichnet. Die Menge der Syndrome, die von der Fehlerklasse F_k (bzw.ω_k) hervorgerufen werden können, sei mit Ω_k bezeichnet. Die Menge Ω aller Syndrome ist also die Vereinigung aller M Syndrommengen Ω_k:

$$\Omega = \bigcup_{k=1}^{M} \Omega_k$$

Eine gute Diagnosevorschrift besteht damit aus einer guten Einteilung von Ω in die M Mengen Ω_k.

4.1 Die Bayes-Strategie

Bei der Diagnose bzw Klasseneinteilung ist es, wie in 2.4, in 2.5 und in 3.4 gezeigt wurde, sinnvoll, auch die damit verbundenen Kosten zu berücksichtigen: die Zahl der Iterationen bei der sequentiellen Diagnose (Zeitkosten) und die Zahl der unnötig ersetzten Einheiten (Materialkosten)(siehe Beispiel 4.1). Das Risiko oder die Kosten, ein Muster $x \varepsilon \Omega_k$ in die Klasse ω_m einzuordnen, sei mit $L_{km}(x,c)$ gegeben; c sei ein Parameter, der die Einordnung der Muster kontrolliert und gibt i.A. die Lage der Klassengrenzen wieder.
Dann sind die erwarteten Kosten R, sich für ein ω_k zu entscheiden, wenn x vorliegt

$$R(\omega_k/x) = \sum_{m=1}^{M} L_{km}(x,c)\, P(\omega_m/x)$$

Die erwarteten Kosten über alle Muster sind dann

$$R(\omega_k) = E\{R(\omega_k/x)\}_x = \sum_{x \varepsilon \Omega} R(\omega_k/x) P(x)$$

Wenn die Ω_m paarweise disjunkt sind gilt

$$R(\omega_k) = \sum_{m=1}^{M} \sum_{x \varepsilon \Omega_m} L_{km}(x,c)\, P(\omega_m/x) P(x)$$

und die Gesamtkosten über alle Klassen

$$R_B := \sum_{k=1}^{M} R(\omega_k)$$

$$= \sum_{k=1}^{M} \sum_{m=1}^{M} \sum_{x \varepsilon \Omega_m} L_{km}(x,c)\, P(\omega_m/x) P(-)$$

Das sog. Bayes-Risiko R_B hängt bei gegebenen L_{km}, $P(x/\omega_k)$ und $P(\omega_k)$ im Wesentlichen von der Unterteilung von Ω in die Untermengen Ω_m ab. Die Unterteilung läßt sich mit einer geigneten Funktion parametrisieren :

$$\theta_m(x,c) := \begin{cases} 0 & x \notin \Omega_m \\ 1 & x \varepsilon \Omega_m \end{cases}$$

somit ist

$$R_B = \sum_{k=1}^{M} \sum_{m=1}^{M} \sum_{x\in\Omega} \theta_m(x,c)\, L_{km}(x,c)\, P(\omega_m/x)P(x)$$

Die Aufgabe einer guten Diagnose ist es, durch Festlegen eines c* die Lage der Klassengrenzen so gut zu wählen, daß

$$R_B(c^*) \stackrel{!}{=} \text{minimal}$$

und mit einem differenzierbaren Minimum

$$\nabla_c R_B(c^*) = 0$$

Wie Tsypkin /TSY/S.61 zeigt, folgen daraus als notwendige Bedingungen

$$\text{(a)} \quad \sum_{k=1}^{M} \sum_{m=1}^{M} \sum_{x\in\Omega} \theta_m(x,c)\{\nabla_c L_{km}(x,c)\}\, P(\omega_m/x)P(x) \Big|_{c=c^*} = 0$$

und für die Muster x, die auf der Grenze zwischen ω_l und ω_m liegen

$$\text{(b)} \quad h_{lm}(x,c^*) := \sum_{k=1}^{M} (L_{kl}(x,c^*) - L_{km}(x,c^*))\, P(x/\omega_k)P(\omega_k) \stackrel{!}{=} 0$$

Die Diskriminanzfunktion h_{lm} läßt sich geometrisch als Hyperfläche zwischen den Klassen ω_l und ω_m interpretieren. Das optimale c* kann man mit (a) bestimmen, während (b) die Entscheidungsregel

$$\text{(4.1a)} \qquad \begin{aligned} h_{lm}(x,c^*) &\leq 0 \quad & x &\in \Omega_l \\ h_{lm}(x,c^*) &> 0 \quad & x &\notin \Omega_l \end{aligned}$$

für die Klasse ω_l die Abgrenzung zu allen anderen Klassen ω_m darstellt.

In der uns interessierenden Anwendung zur Fehlerdiagnose führt eine Verwechselung der Fehlerklasse F_k mit der Fehlerklasse F_m dazu, daß Einheiten, die intakt sind, als defekt angesehen (und repariert) werden und umgekehrt defekte Einheiten nicht erkannt werden. Die Folgen beider Entscheidungen sind einerseits unnötige Reparaturkosten und andererseits zusätzliche Testkosten wenn die Tests wiederholt werden müssen.
Ein Beispiel soll dies verdeutlichen.

BEISPIEL 4.1:
Seien

C_r:= Reparaturkosten für eine unnütz ersetzte Einheit
C_T:= Zeitkosten (z.B.Computerkosten) für eine unnütze Diagnoserunde

Das System bestehe aus den drei Einheiten u_1, u_2 und u_3. Sei $F=\{u_1,u_2\}$ mit dem Syndrom S vorliegend.
Dann gilt:

- Wenn $\sigma(S)=\{u_1,u_2,u_3\}$, so ist u_3 unnötig ersetzt,
  also $L_{F\sigma}= C_r$

- Wenn $\sigma(S)=\{u_1\}$, so ist u_2 nicht ersetzt worden; der Fehler bleibt und u_2 kann erst in einer erneuten Testrunde ersetzt werden:
  $L_{F\sigma}= C_T$

- Wenn $\sigma(S)=\{u_1,u_3\}$, so ist u_3 unnötig und u_2 nicht ersetzt worden. Also ist
  $L_{F\sigma}= C_T+ C_r$

Allgemein läßt sich definieren:

$L_{F\sigma}= C_r\times$ Zahl der zuviel ersetzten Einheiten $+C_T\ \Theta(F,\sigma)$

mit $\Theta(F,\sigma) := \begin{cases} 0 & \text{wenn } \sigma \text{ mind. alle defekten Einheiten ersetzt} \\ 1 & \text{wenn } \sigma \text{ mind. eine defekte Einheit nicht ersetzt} \end{cases}$

Die Höhe der Kosten hängt dabei nur von der falschen Entscheidung, nicht aber vom beobachteten Syndrom oder einer Klassengrenze c ab.
Deshalb läßt sich plausibel annehmen

$$L_{km}= \text{const}_{km} \qquad \text{(klass. Bayes-Risiko)}$$

Damit ist die Bedingung (a) erfüllt.
Mit (b) folgt außerdem

$$h_{lm}(x) := \sum_{k=1}^{M} (L_{kl}-L_{km})P(x/\omega_k)P(\omega_k)$$

mit der Entscheidungsregel

$x \in \omega_l$, Wenn $h_{lm} \leq 0$
$x \notin \omega_l$, Wenn $h_{lm} > 0$

für alle Klassen ω_m.

Für den Fall des 0-1 Loss mit $L_{ii}=0$, $L_{ij}=1$ ist

$$(L_{kl}-L_{km}) \begin{cases} 0 & k=l,\ k=m \\ -1 & k=l,\ k\neq m \\ +1 & k\neq l,\ k=m \\ 0 & k\neq l,\ k\neq m \end{cases}$$

und damit

$$h_{lm} = -P(x/\omega_l)P(\omega_l) + P(x/\omega_m)P(\omega_m)$$

und die Diagnoseregel ist

(4.1b) $x \in \omega_l$, wenn für alle ω_m

$$P(x/\omega_m)P(\omega_m) \leq P(x/\omega l\ )P(\omega_l)$$

gilt.

Eine Diskussion der Bayes-Diagnose für die Fehlerdiagnose ist auch in /DAL2/ zu finden.

Im Folgenden sollen verschiedene Wahrscheinlichkeitsdiagnosestrategien miteinander verglichen werden.

4.3 Maximum-Likelihood Diagnosestrategie

Diese Strategie entscheidet für die Fehlerklasse, die das Syndrom x am wahrscheinlichsten produziert hat:

$$\text{Wähle } \omega_l \text{ mit } P(x/\omega_l) = \max_m P(x/\omega_m)$$

Wenn die Auftrittswahrscheinlichkeit der Klassen nicht gleich ist, müssen die bedingten Wahrscheinlichkeiten noch gewichtet werden und die Strategie wird zu

$$\text{Wähle } \omega_l \text{ mit } P(x/\omega_l)P(\omega_l) = \max_m P(x/\omega_m)P(\omega_m)$$

4.3 Maximum a-posteriori Diagnosestrategie

Diese Strategie versucht bei der Diagnose die Klasse auszuwählen, zu der x am wahrscheinlichsten gehört:

$$\text{Wähle } \omega_1 \text{ mit } P(\omega_1/x) = \max_m P(\omega_m/x)$$

Da P(x) nicht von der Klassenwahl abhängt, gilt für ω_1 mit P(a/b)P(b)=P(b/a)P(a)

$$P(\omega_1/x)P(x) = \max_m P(\omega_m/x)P(x)$$

$$= \max_m P(x/\omega_m)P(\omega_m)$$

die selbe Auswahlregel wie bei der maximum likelihood Diagnose. Mit dem 0-1 Loss zeigt sich in (4.1b), daß beide Strategien nur Spezialfälle der allgemeinen Bayesstrategie sind.

4.4 Die Siegert-Kotelnikov Diagnosestrategie

Die Minimierung der Zahl der falschen Entscheidungen ist das Ziel dieser Strategie. Dazu wird die Wahrscheinlichkeit der falschen Entscheidungen

$$R_S := \sum_{k=1}^{M} \sum_{x \notin \Omega_k} P(x/\omega_k)P(\omega_k)$$

aufgestellt. Beim Vergleich mit R_B ergibt sich

$$R_S = \sum_{k=1}^{M} \sum_{m=1}^{M} \sum_{x \in \Omega_m} L_{km}(x,c)\, P(\omega_m/x)P(x) = R_B$$

bei den speziellen Kosten

$$L_{km} := \begin{cases} 0 & k=m \\ 1 & k \neq m \end{cases}$$

Die Strategie, die Zahl der falschen Entscheidungen so klein wie möglich zu machen, ist also auch wieder äquivalent mit der Minimisierung eines speziellen Bayes-Risikos.

Äquivalent zu der obigen Diagnosestrategie ist auch die Strategie von Blount (vgl Abschnitt 3.3), die Zahl der richtigen Entscheidungen so groß wie möglich zu machen. Das Maximum von D_{sys}

$$D_{sys} = \sum_{k=1}^{M} \sum_{x \in \Omega_k} P(x/\omega_k)P(\omega_k)$$

$$= \sum_{k=1}^{M} \{1 - \sum_{x \notin \Omega_k} P(x/\omega_k)P(\omega_k)\}$$

ist gegeben, wenn der Klammerausdruck minimal wird. Dies ist gleichbedeutend mit der Minimisierung von

$$R_{sys} := \sum_{k=1}^{M} \sum_{x \notin \Omega_k} P(x/\omega_k)P(\omega_k) = R_S$$

Die beiden Strategien haben damit auch gleiche Entscheidungsregeln.

4.5 Die gemischte Diagnosestrategie

Die gemischte Diagnosestrategie ('Mixed Decision') versucht, die Differenz zwischen der Zahl der richtigen und der Zahl der falschen Entscheidungen zu maximieren:

$$P(x \text{ richtig eingeordnet}) - L \cdot P(x \text{ falsch eingeordnet}) \stackrel{!}{=} \max$$

mit der Gewichtung L.
Es ist

$$P(x \text{ richtig in } \omega_i \text{ eingeordnet}) = \sum_{x \in \Omega_i} P(x/\omega_i)P(\omega_i)$$

$$P(x \text{ falsch in } \omega_i \text{ eingeordnet}) = \sum_{k=1}^{M} \sum_{x \in \Omega_i} P(x/\omega_k)P(\omega_k)$$

Die Gesamtsumme über alle Gebiete Ω_i ist somit

$$\sum_{i=1}^{M} \{ \sum_{x\varepsilon\Omega_i} P(x/\omega_i)P(\omega_i) - L \sum_{k=1}^{M} \sum_{x\varepsilon\Omega_i} P(x/\omega_k)P(\omega_k) \}_{i\neq k}$$

$$= \sum_{i=1}^{M} \{ 1 - \sum_{x\varepsilon\Omega_k} P(x/\omega_k)P(\omega_k) - L \sum_{k=1}^{M} \sum_{x\varepsilon\Omega_i} P(x/\omega_k)P(\omega_k) \}_{i\neq k}$$

Dies ist maximal, wenn

$$R_M := \sum_{i=1}^{M} \sum_{k=1}^{M} \{ \sum_{x\varepsilon\Omega_k} P(x/\omega_i)P(\omega_i) + L \sum_{x\varepsilon\Omega_i} P(x/\omega_k)P(\omega_k) \}_{i\neq k}$$

minimal wird. Da

$$\sum_{i=1}^{M} \sum_{k=1}^{M} \sum_{x\varepsilon\Omega_k} P(x/\omega_i)P(\omega_i) = \sum_{i=1}^{M} \sum_{k=1}^{M} \sum_{x\varepsilon\Omega_i} P(x/\omega_k)P(\omega_k)_{i\neq k}$$

ist

$$R_M = \sum_{\substack{i=1 \\ i\neq k}}^{M} \sum_{k=1}^{M} \sum_{x\varepsilon\Omega_k} (1+L)\, P(x/\omega_i)P(\omega_i)$$

als spezielles Bayes-Risiko erkennbar mit

$$L_{ik} \begin{cases} 0 & i=k \\ (1+L) & i\neq k \end{cases}$$

4.6 Die Minimax Strategie

Die Minimax-Diagnosestrategie versucht bei fehlenden Informationen über die a-priori- Wahrscheinlichkeiten $P(\omega_i)$ den Schaden für die Diagnose dadurch zu begrenzen, daß unter der Annahme schlechtester Voraussetzungen

$$P_i := P(\omega_i) \quad \text{mit} \quad R_B(P_i) = \max_j R_B(P(\omega_j))$$

das Minimum davon bei der Klasseneinteilung gesucht wird.
Angenommen, diese M Zahlen P_i sind bestimmt worden.
Dann wird das Risiko

$$R_{MM} := \sum_{k=1}^{M} \sum_{m=1}^{M} \sum_{x \in \Omega_m} L_{km} \, P(x/\omega_k) \, P_k$$

bei folgender Diskriminanzfunktion minimal

$$h_{lm} := \sum_{k=1}^{M} (L_{kl} - L_{km}) P(x/\omega_k) P_k$$

Zur Anwendung der Minimax-Strategie in verteilten Entscheidungs-Systemen siehe /SAN/.

5.0 Vergleich der Ansätze

Im vorigen Abschnitt wurden verschiedene wahrscheinlichkeitsdiagnostische Ansätze untersucht und miteinander verglichen. Dabei stellte sich heraus, daß die verschiedenen Ansätze spezielle Versionen der Bayes-Strategie darstellen. In diesem Abschnitt soll die Leistungsfähigkeit dieser universellen Diagnosestrategie mit den Diagnoseansätzen der Abschnitte 2 und 3 verglichen werden, die im Folgenden nochmals kurz zusammengefaßt werden.

5.1 Prob. Diagnose und t-Fehler Diagnostizierbarkeit ohne Reparatur

Sei ein Syndrom in System mit einem Testgraphen nach 2.1 gegeben. Dann läßt sich eine deterministische Diagnose definieren:

> Wähle diejenige Fehlerklasse, die zum Syndrom konsistent ist und die kleinste Zahl von Fehlern hat.

Wenn es mehrere Fehlerklassen mit gleicher Fehlerzahl gibt, treten Probleme auf. Deshalb soll das betrachtete System eingeschränkt werden.
Sei in einem System mit t-diagnostizierbarem Testgraphen (ohne Reparatur) Ω die Menge aller Syndrome und $\Omega_1(t) \subset \Omega$ die Menge aller Syndrome, die von Fehlerklassen mit $\leq t$ defekten Einheiten hervorgebracht werden. Somit gibt es nach Voraussetzung für jedes Syndrom $S \in \Omega_1(t)$ nur eine Fehlerklasse mit $\leq t$ defekten Einheiten und es läßt sich eine deterministische Diagnose definieren, bei der für jedes Syndrom die dazugehörige Fehlerklasse aus einer festen, von der Zuverlässigkeit der Einheiten unabhängigen Liste herausgesucht wird:

(5.1a) Wähle für jedes Syndrom $S \subset \Omega_1(t)$ die eine Fehlerklasse $\sigma_{det}(S)$, die S hervorbringt und $\leq t$ fehlerhafte Einheiten hat.

Damit ist eine Diagnose für alle $S \in \Omega_1(t)$, nicht aber für $S \in \Omega_2$ definiert. Dies läßt sich durch eine pessimistische Diagnose ('wenn $S \notin \Omega_1(t)$, so sind alle Einheiten des Systems defekt') oder eine Wahrscheinlichkeitsdiagnose nach (5.1c) vornehmen.

Die Diagnose nach Maheshwari und Hakimi (s.3.1) sei

(5.1b) Wähle für jedes Syndrom S eine Fehlerklasse $\sigma_{Mah}(S)$, die S

hervorbringt und für die

$P(\sigma_{Mah}) = \max_F P(F)$, F konsistent zu S, ist.

In p-T diagnostizierbaren Systemen (vgl.Abschnitt 3.1) gibt es für jedes Syndrom jeweils nur eine Fehlerklasse, für die $P(F) > T$ gilt und damit das einzige Maximum bildet.

Die Diagnosevorschrift von Blount (s.Abschnitt 3.3) ist

(5.1c) Wähle für jedes Syndrom S eine Fehlerklasse $\sigma_{Blo}(S)$, für die $P(S,\sigma_{Blo}(S)) \stackrel{!}{=} \max_F P(S,F)$ gilt.

Die Wahrscheinlichkeitsdiagnose nach der Bayes-Strategie ist nach Abschnitt 4.1

(5.1d) Wähle für jedes Syndrom S eine Fehlerklasse $\sigma_{Bay}(S)$, so daß für alle anderen Fehlerklassen F_k gilt $h_{ik}(S) \leqq 0$ mit $F_i = \sigma_{Bay}(S)$.

Wie in Abschnitt 4.4 gezeigt worden ist, sind die Diagnosestrategien 5.1c von Blount und die Bayes-Diagnose 5.1d identisch, wenn für die Kostenkoeffizienten L_{ij} das 0-1 Loss gewählt wird. Damit ist 5.1c ein Spezialfall von 5.1d.

Was sind nun die typischen Gemeinsamkeiten und Unterschiede der Diagnosen ? Bei allgemeinen Kostenkoeffizienten entscheidet die Bayes-Diagnose 5.1d sicher anders als die restlichen drei Diagnosestrategien, die keinerlei Kostenkalkulationen enthalten.

Um die näheren Zusammenhänge der Diagnosen (und damit auch der dahinterstehenden Modelle) zu untersuchen, soll deshalb im Folgenden nur der Spezialfall der Bayes-Diagnose mit 0-1 Loss 5.1c im Verhältnis zu den Diagnosen 5.1a und 5.1b betrachtet werden.

Dazu wird folgender Hilfssatz benötigt:

LEMMA 5.1

Seien N unabhängige, gleiche Einheiten mit der Zuverlässigkeit R>0,5 gegeben und es ist $S \varepsilon \Omega_1(t)$. Dann gilt:

Jede Diagnose von S in t-diagnostizierbaren Testgraphen ohne Reparatur entscheidet für die gleiche Fehlerklasse wie die deterministische Diagnose d.u.n.d., wenn sie die zum Syndrom S kompatible Fehlerklasse mit der größten Auftrittswahrscheinlichkeit wählt.

BEWEIS:

a) notwendige Bedingung:

Sei F_i die einzige zum Syndrom kompatible Fehlerklasse mit $|F_i| \leq t$ defekten Einheiten, die nach Voraussetzung in t-diagnostizierbaren Graphen ohne Reparatur, für die die Diagnose 5.1a definiert ist, existieren muß. Dann wird von der deterministischen Diagnose auf F_i erkannt. Sei F_j eine weitere, zum Syndrom kompatible Fehlerklasse. Es ist $|F_j| > t$ und

$$P(F_i) = (1-R)^{|F_i|} R^{N-|F_i|}$$

$$= (1-R)^{|F_i|-|F_j|} (1-R)^{|F_j|} R^{N-|F_j|} R^{|F_j|-|F_i|}$$

$$= P(F_j) \left\{\frac{R}{1-R}\right\}^{|F_j|-|F_i|}$$

Mit

$$R > 0{,}5 \quad \Rightarrow \quad \frac{R}{1-R} > 1 \quad \text{Wenn } |F_j|-|F_i| > 0$$

ist

$$P(F_i) > P(F_j) \ .$$

Da dies für alle F_j gilt, wählt jede Diagnose, die für F_i entscheidet, die Fehlerklasse mit der größten Auftrittswahrscheinlichkeit.

b) hinreichende Bedingung:

Sei von einer nichtdeterministischen Diagnose die Fehlerklasse F_i gewählt als diejenige mit der größten Auftrittswahrscheinlichkeit.

$$P(F_i) = \max_j P(F_j) \qquad F_j \text{ kompatibel zum Syndrom}$$

Dann ist mit obiger Rechnung $|F_j|-|F_i| \geq 0$ und somit $|F_i| \leq |F_j|$, wobei $F_j \varepsilon \{F/ F$ kompatibel zu $S\}$ ist. Da es nach Voraussetzung für alle $S \varepsilon \Omega_1(t)$ jeweils nur eine Fehlerklasse mit $|F_k| \leq t$ defekten Einheiten gibt, ist diese Fehlerklasse identisch mit der Fehlerklasse F_i und F_i wird ebenfalls von der deterministischen Diagnose (5.1a) erwählt. Q.E.D.

Damit läßt sich nun der Zusammenhang der Diagnosen (5.1a) und (5.1b) formulieren:

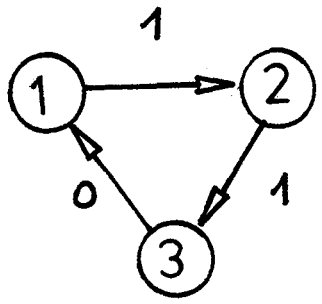
SATZ 5.1

Seien N unabhängige, gleiche Einheiten mit $R > 0.5$ gegeben. Dann diagnostizieren in t-diagnostizierbaren Testgraphen ohne Reparatur die probabilistische Diagnose und die deterministische Diagnose für alle $S \varepsilon \Omega_1(t)$ gleich.

BEWEIS:

Nach Definition wählt die probabilistische Diagnose (5.1b) immer die Fehlerklasse mit der größten Auftrittswahrscheinlichkeit, die kompatibel zum vorliegenden Syndrom ist. Nach Lemma 5.1 ist dies die selbe Fehlerklasse die auch die deterministische Diagnose wählt. Q.E.D.

Der Satz gilt, wenn alle Einheiten gleiche Zuverlässigkeiten R_i aufweisen. Sind sie dagegen verschieden, so sind probabilistische und deterministische Diagnosen im allgemeinen nicht identisch. Dies sei in folgendem Beispiel verdeutlicht.



t=1

$R_1 := R_3 := 0.6$, $R_2 := 0.9$

$F_1 := \{u_1, u_3\}$ $\quad P(F_1) = 0.144$

$F_2 := \{u_2\}$ $\quad P(F_2) = 0.036$

Abb.5.1a Beispiel für ungleiche Diagnosen

Die deterministische Diagnose (5.1a) entscheidet auf F_2, die probabilistische Diagnose (5.1b) auf F_1, obwohl $R_i > 0.5$ ist.

Für die Diagnose (5.1a) und (5.1c) gilt folgender Satz:

SATZ 5.2

Seien N unabhängige,gleiche Einheiten mit R>0.5 gegeben. Außerdem seien die Modellannahmen von Preparata (s. Tabelle 1 in 4.1) gültig. Dann gilt folgender Satz:
Die deterministische Diagnose (5.1a) und die probabilistische Diagnose (5.1c) entscheiden in t-diagnostizierbaren Testgraphen (ohne Reparatur) gleich d.u.n.d., wenn

a) jede Einheit die gleiche Zahl von Tests ausführt (d_{out}=const)

b) p = 0,5

BEWEIS: (hinreichende Bedingung)

Sei F_i die zum Syndrom konsistente Fehlerklasse der deterministischen Diagnose. Für jede andere konsistente Fehlerklasse F_j gilt $|F_i|<|F_j|$. Mit allen M:=|E| Tests t^k_{mn}, k=1..M zwischen je zwei Einheiten $u_m,u_n \varepsilon V$ im Zustand e_m,e_n mit

mit $e_i := \begin{cases} 0 & u_i \text{ intakt} \\ 1 & u_i \text{ defekt} \end{cases}$

gilt

$$P(S/F_i) = \prod_{k:=1}^{M} P(t^k_{mn}/(e_m e_n))$$

Seien n_0 testende Einheiten intakt, n_1 defekt mit Testergebnis $t^k_{mn}=1$ und n_2 defekt mit $t^k_{mn}=0$. Das Ereignis $t^k_{mn}=x$ bedeutet, daß der Test t^k_{mn} das Ergebnis x (0 oder 1) hat.
Somit ergibt sich

$$P(S/F_i) = P(t^k_{mn}=x/(0x))^{n_0} P(t^k_{mn}=1/(1x))^{n_1} P(t^k_{mn}=0/(1x))^{n_2}$$

mit $x \varepsilon \{0,1\}$ als Zustand einer getesteten Einheit.

Nach Voraussetzung 2.1a,b ist

$$P(t^k_{mn}=x/(0x))=1, \quad P(t^k_{mn}=1/(1x))=1-p, \quad P(t^k_{mn}=0/(1x))=p.$$

Bei konstantem d_{out} und p=1/2 ist

$$n_0=(N-|F_i|)d_{out} \quad n_1+n_2=|F_i|d_{out}$$

und somit

$$P(S/F_i) = (1/2)^{d_{out}|F_i|}$$

Für jede andere Fehlerklasse F_j ist

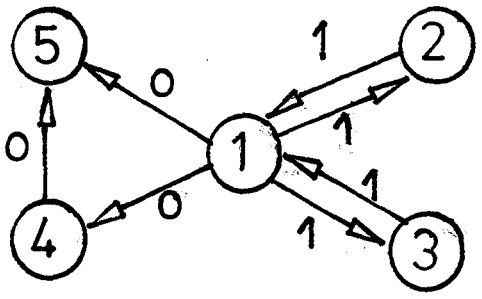
$$P(S/F_j)=(1/2)^{d_{out}|F_j|} < (1/2)^{d_{out}|F_i|} = P(S/F_i)$$

Also wählt die probabilistische Diagnose (5.1c) ebenfalls F_i, da mit Lemma 5.1 $P(F_i)>P(F_j)$ und damit

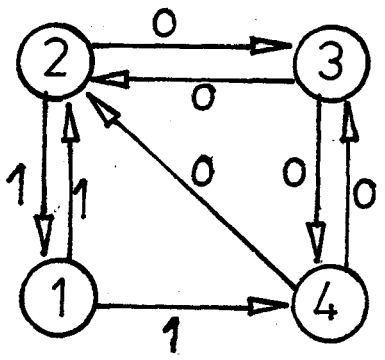
$$P(S/F_i)P(F_i) > P(S/F_j)P(F_j)$$

bei allen anderen Fehlerklassen ist.

Um auch die Notwendigkeit der Bedingungen zu zeigen, reicht es, Gegenbeispiele anzugeben, bei denen eine der beiden Bedingungen nicht gegeben ist und bei denen unterschiedliche Diagnose erfolgt.



b)



c)

Abb. 5.1b,c Gegenbeispiele

a) Sei d_{out} nicht konstant und $p<0.57$, also $p\neq 1/2$ oder $p=1/2$.
Der Testgraph aus Abb. 5.1b hat $t\geq 1$, da wenn Einheiten 4 oder 5 defekt sind, dies direkt von Einheit 1 festgestellt wird. Ist Einheit 1 defekt, so wird dies von Einheiten 2,3 festgestellt. Ist Einheit 2 oder 3 defekt, so wird dies durch Übereinstimmung von 1,2 oder 1,3 diagnostiziert. Außerdem ist $t<2$, da der Fehler $F=\{u_1,u_4\}$ nicht von $F'=\{u_1\}$ unterschieden werden kann (s. Bedingung 2.1d).

Also ist t=1 und die deterministische Diagnose erwählt für das vorliegende Syndrom $F_1=\{u_1\}$ mit

$$P(S/F_1)P(F_1)=(1-R)R^4(1-p)^2\ p^2,$$

während die probabilistische Diagnose $F_2=\{u_2,u_3\}$ erwählt mit

$$P(S/F_2)P(F_2)=(1-R)^2\ R^3(1-p)^2,$$

wenn mit $R:=0.6$ $p^2<(1-R)/R=1/3$ und damit $p\leq 0.577$ ist.

b) Sei d_{out}=const und $p\neq 1/2$.
Der Testgraph aus Abb.5.1c hat $t\geq 1$ wegen der eingebetteten Ringstruktur und $t<2$, da Einheit 1 nur von einer Einheit getestet wird (Bedingungen 2.1c,d).
Sei ein Syndrom gegeben wie eingezeichnet und $F_1=\{u_1\}$, $F_2=\{u_2,u_3,u_4\}$, $R=0.6$, $p=0.9$.
Dann erwählt die deterministische Diagnose F_1, während die probabilistische Diagnose mit

$$P(S,F_1)=(1-p)^2(1-R)R^3=0.864\ 10^{-3}$$

und $P(S,F_2)=(1-p)p^5(1-R)^3\ R=2.26\ 10^{-3}$

für die Fehlerklasse F_2 entscheidet.

Für optimale t-diagnostizierbare Testgraphen ohne Reparatur läßt sich sogar ein stärkerer Satz formulieren, jedoch nur der hinreichende Teil:

SATZ 5.3:

In D_{1t}-Graphen mit gleichen, unabhängigen Einheiten entscheiden die deterministische Diagnose (5.1a) und die probabilistische Diagnose (5.1c) für alle Syndrome $S_{\varepsilon\Omega_1}(t)$ gleich, wenn folgende Voraussetzungen gegeben sind:

a) $p=1/2$
b) $R > 1/(2^t+1)$

BEWEIS:

Sei F_i diejenige Fehlerklasse, für die die deterministische Diagnose entscheidet und F_j eine andere zum Syndrom kompatible Fehlerklasse. Für diese gilt $|F_j|\geq|F_i|$. Nach dem Beweis von Satz 5.2 ist bei $p=0.5$

$$P(S/F_i) = (1/2)^{d_{out}\,|F_i|}$$

Da in D_{1t}-Graphen $d_{out}=t$ ist, gilt

$$P(S/F_i)P(F_i) = (1/2)^{t|F_i|}\ (1-R)^{|F_i|}\ R^{N-|F_i|}$$

Also gilt

$$\begin{aligned}
&P(S/F_j)P(F_j)\\
&= (1/2)^{t|F_j|}\ (1-R)^{|F_j|}\ R^{N-|F_j|}\\
&= (1/2)^{t(|F_j|-|F_i|)}\ (1/2)^{t|F_i|}\ (1-R)^{|F_j|-|F_i|}\ (1-R)^{|F_i|}\ R^{N-|F_i|}\\
&\qquad *R^{|F_i|-|F_j|}\\
&= (1/2)^{t(|F_j|-|F_i|)}\ (1-R)^{|F_j|-|F_i|}\ R^{|F_i|-|F_j|}\ P(S/F_i)P(F_i)\\
&= \{(1/2)^{t}\ (1-R)/R\}^{|F_j|-|F_i|}\ P(S/F_i)P(F_i)
\end{aligned}$$

Nach Voraussetzung ist

$$\begin{aligned}
R > 1/(2^t+1) \quad &\Rightarrow \quad 2^tR + R > 1\\
&\qquad\quad 2^t > (1-R)/R\\
&\qquad\quad 1 > \frac{1}{2^t}\ \frac{1-R}{R}
\end{aligned}$$

Da $|F_j|-|F_i|>0$ ist damit auch

$$\{\ (1/2)^{t}\ (1-R)/R\ \}^{|F_j|-|F_i|} < 1$$

und somit ist

$$P(S/F_j)P(F_j) < P(S/F_i)P(F_i)$$

so daß auch die probabilistische Diagnose (5.1c) auf F_i entscheidet
Q.E.D.

Bisher wurden in Satz 5.1, Satz 5.2 und Satz 5.3 Bedingungen angegeben, unter denen die deterministische Diagnose (5.1a) mit den probabilistischen Diagnosen (5.1b) und (5.1c) übereinstimmen. Im Folgenden sollen die Unterschiede zwischen den Diagnosen untersucht werden.
Dazu benötigen wir ein Maß für die Fähigkeit einer Diagnose σ, die dem beobachteten Syndrom S_1 zugrunde liegende Fehlerklasse F_i richtig zu

erkennen. Sei die 'Diagnostizierbarkeit D_σ', die Wahrscheinlichkeit einer richtigen Diagnose, ein solches Maß (vgl./DAL/ S.62 und /DAL3/).
Dies ist die Wahrscheinlichkeit, daß die Diagnose σ bei allen auftretenden Syndromen S_1 auf die dem jeweiligen Syndrom zugrunde liegende Fehlerklasse F diagnostiert:

$$D_\sigma := \sum_1 P(\sigma(S_1)=F, S_1(F))$$
$$= \sum_1 P(\sigma(S_1)/S_1)P(S_1)$$

Sei der Diagnosebereich der deterministischen Diagnose auf ganz Ω, wie bei (5.1a) angedeutet, erweitert. Für die Diagnostizierbarkeit eines Systems unter Verwendung der verschiedenen Diagnosen gilt

SATZ 5.4:

Die Diagnostizierbarkeit eines Systems ist unter der probabilistischen Diagnose (5.1c) immer größer oder gleich als unter jeder anderen, also auch der deterministischen, Diagnose.

BEWEIS:

Sei eine andere Diagnose gegeben. Diese Diagnose weise dem Syndrom S_1 die Fehlerklasse $\sigma_x(S_1)$ zu. Da die probabilistische Diagnose $\sigma_{prob}(S_1)$ so sucht, daß

$$P(S_1/\sigma_{prob}(S_1))P(\sigma_{prob}(S_1)) \geq P(S_1/F_i)P(F_i)$$

für jede Fehlerklasse F_i im System gilt, ist

$$P(S_1/\sigma_{prob}(S_1))P(\sigma_{prob}(S_1)) \geq P(S_1/\sigma_x(S_1))P(\sigma_x(S_1))$$

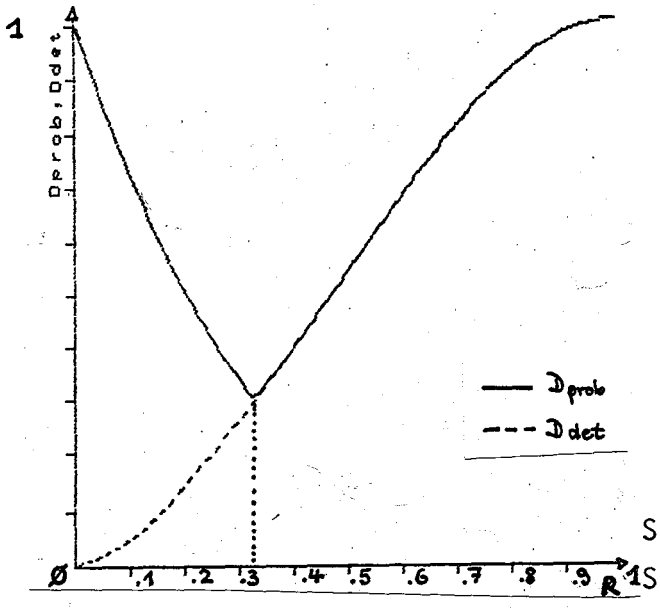
und somit

$$D_{\sigma_{prob}} = \sum_1 P(S_1/\sigma_{prob}(S_1))P(\sigma_{prob}(S_1))$$
$$\geq \sum_1 P(S_1/\sigma_x(S_1))P(\sigma_x(S_1)) = D_{\sigma x}$$

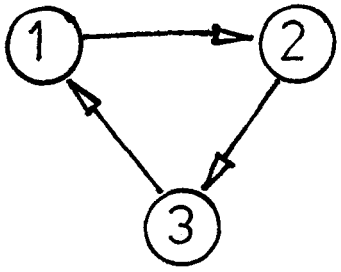
Q.E.D.

Satz 5.4 ergibt sich durch die Wahl des Vergleichsmaßes 'Diagnostizierbarkeit' als der Wahrscheinlichkeit, richtig zu diagnostizieren. Dies ist identisch mit der Forderung von Blount (s. Abschnitte 3.3 und 4.4), die Zahl der richtigen Entscheidungen zu maximieren. Die Diagnosevorschrift (5.1c), die D_{sys}

maximiert, ist somit auch optimal für D_σ.
Die Unterschiede der Diagnosen soll an einem einfachen Beispiel, dem D_{1t} Graphen mit t=1, erläutert werden.



Diagnostizierbarkeit



D_{11} Testgraph

S_0=(000) F_0=(000) S_4=(100) F_4=(100)
S_1=(001) F_1=(001) S_5=(101) F_5=(101)
S_2=(010) F_2=(010) S_6=(110) F_6=(110)
S_3=(011) F_3=(011) S_7=(111) F_7=(111)

Abb 5.1d Diagnosebeispiel

Mit seinen 3 Einheiten sind 2^3=8 Fehlerklassen und bei 3 Tests ebenfalls 8 Syndrome möglich; die Syndrommenge $\Omega_1(t)$ ist $\Omega-\{S_7\}$.
In Abb.5.1d ist die Diagnostizierbarkeit des Systems bei deterministischer und probabilistischer Diagnose in Abhängigkeit von der Zuverlässigkeit R der statistisch unabhängigen Einheiten aufgetragen.
Mit t=1 kann die deterministische Diagnose nur dann immer korrekt diagnostizieren, wenn nicht mehr als eine Einheit ausgefallen ist. Wie zu erwarten nimmt bei absinkendem R die Auftrittswahrscheinlichkeit der Syndrome zu, die zwar in $\Omega_1(t)$ sind, aber von einer Fehlerklasse mit mehr als einer defekten Einheit stammen.
Wie Satz 5.3 zeigt, wird die Diagnose und damit auch die Diagnostizierbarkeit D_σ in diesem Beispiel bei $R \geqq 1/3$ durch den probabilistischen Ansatz nicht verbessert; beide Diagnose sehen die Einheit u_m mit t_{nm}=1 und t_{1n}=0 primär als defekt an: $\sigma(S_1)=F_4$, $\sigma(S_2)=F_1$, $\sigma(S_4)=F_2$, $\sigma(S_3)=F_1$, $\sigma(S_5)=F_4$, $\sigma(S_6)=F_2$.
Anders ist dies bei $R<1/3$. Hier nutzt die probabilistische Diagnose die Information über R aus, entscheidet immer für F_7 und wird damit deutlich

besser als σ_{det}.
Einer der wichtigsten Unterschiede ist also die Beschränkung der deterministischen Diagnose (5.1a) auf die Annahme von $\leqq t$ Fehlern im System. Wie groß ist unter dieser Voraussetzung die Diagnostizierbarkeit im System? Die Wahrscheinlichkeit der richtigen Diagnose, vorausgesetzt $\leqq r$ Einheiten sind defekt, ist

$$D_{\sigma,r} := \frac{\sum\limits_{\substack{l \\ |\sigma(S_l)| \leq r}} P(\sigma(S_l)/S_l)P(S_l)}{\sum\limits_{\substack{i \\ |F_i| \leq r}} P(F_i)}$$

Betrachten wir nun wieder unser voriges Beispiel 5.1a. Für r=0,1,2,3 ist $D_{\sigma,r}$ in Abb. 5.1e aufgetragen. Wie nicht anders zu erwarten, ist für r=0,1 die bedingte Diagnostizierbarkeit gleich 1, also immer 100 prozentig. Anders verhält es sich dagegen, wenn realistischere, weniger restriktive Annahmen über die maximale Zahl von ausgefallenen Einheiten gemacht werden wie r=2,3. Besonders bei dem kritischen Wert der Zuverlässigkeit R=1/3 ist der Unterschied besonders groß: Wird die unbegründete Annahme von maximal einer defekten Einheit fallengelassen, so zeigt sich, daß an Stelle einer 100 prozentigen Diagnose nur knapp 30÷ aller Fälle richtig diagnostiziert werden.

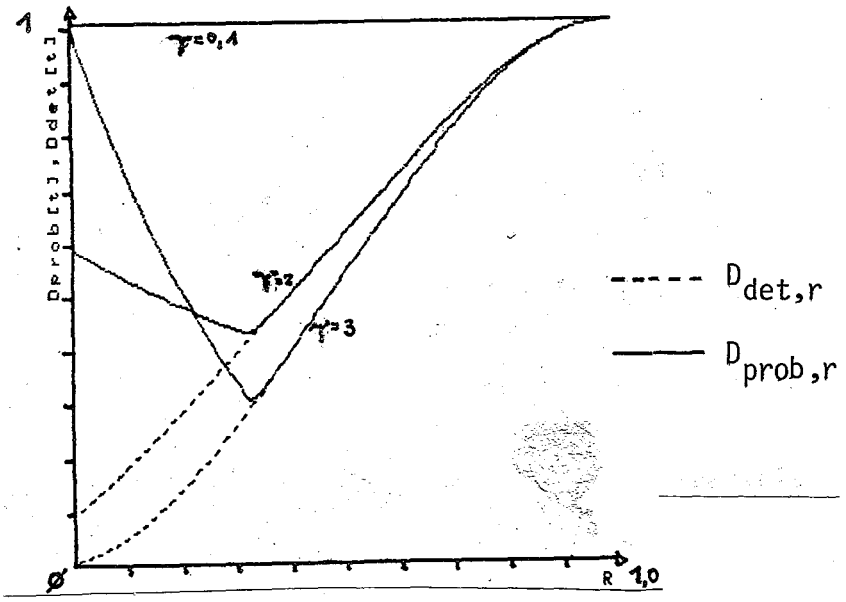


Abb. 5.1e bedingte Diagnostizierbarkeit

5.2 Diagnose nach der Bayesstrategie

Da die Bayesdiagnose bei der Wahl der Fehlerklasse außer dem beobachteten Syndrom S und den zu Grunde liegenden Überlebenswahrscheinlichkeiten R_i der Einheiten auch die bei der Diagnose entstehenden Kosten berücksichtigt, soll sie im folgenden Abschnitt gesondert behandelt werden.
Wie in Abschnitt 4.1 hergeleitet wurde, wird bei der Bayesdiagnose die Fehlerklasse F_m mit dem kleinsten Risiko $r_m(S)$ gewählt, also

$$r_m(S) = \sum_{k:=1}^{2^N} L_{km} P(S/F_k) P(F_k) = \text{minimal}$$

Wie in 4.1 gezeigt wurde, reduziert sich unter der speziellen Annahme des 0-1 Loss die Bayesdiagnose 5.1d zur probabilistischen Diagnose 5.1c. Wie verhält sich nun die Bayesdiagnose, wenn diese spezielle Annahme nicht gemacht wird? Dazu betrachten wir den Risikokoeffizienten aus dem Beispiel aus 4.1:

L_{km}:= Risiko, für F_m zu entscheiden statt für F_k
$= C_r |F_m - F_k| + C_T \theta_{km}$

mit der Kardinalität der Mengendifferenz

$|F_m - F_k|$ = Zahl der mit F_m gegenüber F_k zuviel ersetzten Einheiten

und

$$\theta_{km} := \begin{cases} 0 & \text{,wenn mindestens alle Einheiten aus } F_k \text{ auch in } F_m \text{ sind: } F_m \subset F_k \\ 1 & \text{,wenn mindestens eine Einheit aus } F_k \text{ nicht in } F_m \text{ ist: } F_k - F_m \neq \emptyset \end{cases}$$

und

C_r := Kosten, um eine Einheit zu reparieren
C_T := Kosten für eine Diagnose, die anfällt, wenn eine defekte Einheit nicht repariert wird und damit eine neue Diagnoserunde nötig wird.

Die erwarteten Kosten unter einer beliebigen Diagnose σ_x sind bei dem Syndrom S und der diagnostizierten Fehlerklasse $\sigma_x(S)$

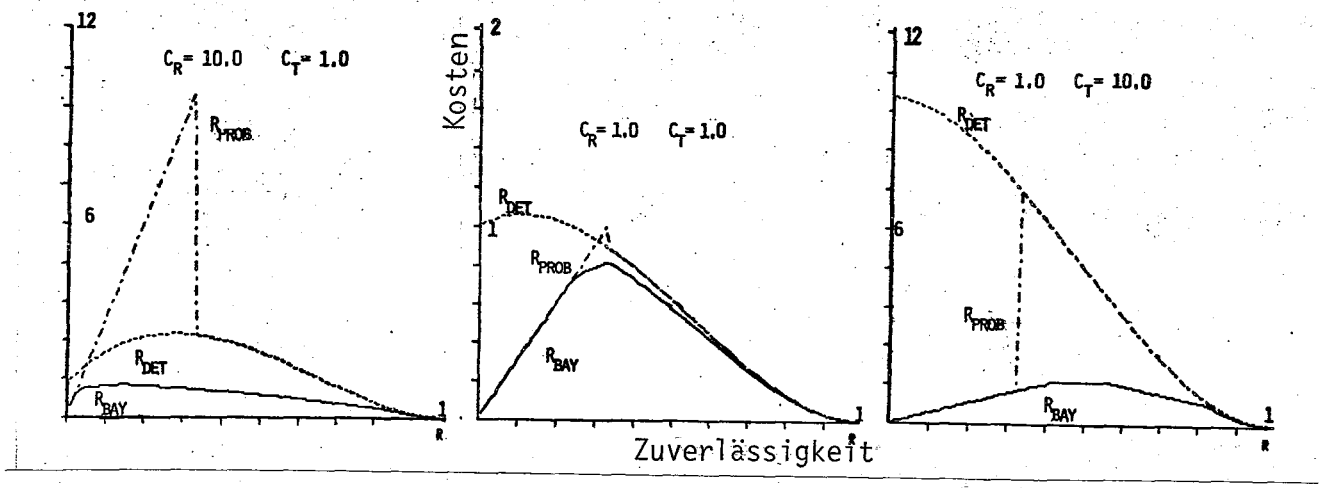
$$r_\sigma(S) = \sum_{k:=1}^{2^N} L_{k\sigma_x(S)} P(S/F_k) P(F_k)$$

und über alle Syndrome

$$R_\sigma := \sum_S r_\sigma(S)$$

$$= \sum_{l:=1}^{2^M} \sum_{k:=1}^{2^N} L_{k\sigma(S)} P(S/F_k) P(F_k)$$

Die Kosten R_σ der verschiedenen Diagnosen sollen an dem einfachen Beispiel des D_{1t}-Graphen mit t=1, N=3, p=1/2 aus Abschnitt 5.1 Abb.5.1a untersucht werden. Die folgenden Abbildungen zeigen die Kosten der deterministischen, probabilistischen und der Bayesdiagnose in Abhängigkeit der Zuverlässigkeit R der Einheiten bei verschiedenen Parameterwerten von C_r und C_T.



a)C_r/C_T=10 b)C_r/C_T=1 c)C_r/C_T=0.1

Abb. 5.2a,b,c Die Kosten der Diagnosearten beim D_{11} Graphen

Wie nicht anders zu erwarten, verursacht die kostengünstige Bayesdiagnose in allen drei Fällen immer gleiche oder geringere Kosten als die beiden anderen Diagnosestrategien.

Da bei R>1/3 die deterministische und die probabilistische Diagnose nach Satz 3 gleich sind, sind auch die Kosten gleich, aber höher als bei σ_{Bay}. Das letztere ist auf eine bessere Anpassung von σ_{Bay} an die gegebene Kostensituation zurückzuführen. Entscheidet beispielsweise bei C_r=1, C_T=10, R=0.75 die deterministische Diagnose $\sigma_{det}(S_1)$ auf F_4, so wählt dagegen $\sigma_{Bay}(S_1)$ die Fehlerklasse F_6 und ersetzt damit nicht nur die vermutlich defekte Einheit u_1, sondern auch u_2, da es bei der unbrauchbaren Testaussage t_{12} billiger ist, gleich u_2 zu ersetzen als eine Diagnoserunde

abzuwarten.
Bei $C_r \sim C_T$ verhält sich die Bayes-Diagnose fast ebenso wie die probabilistische Diagnose. Erst bei $C_r \neq C_T$ zeigen sich die typischen Unterschiede. Die von der probabilistischen Daignose bei $R<1/3$ angewandte Strategie, beim Auftreten eines Syndroms $S\neq(000)$ gleich alle Einheiten auszutauschen, ist bei billigen Einheiten eine gute Lösung, die σ_{Bay} in Bild c) auch bei $R=0.5$ anwendet. Da mit oftmals ungenügenden Reparaturen auch teure erneute Diagnosen nötig sind, verursacht σ_{det} in diesem Fall die höchsten Kosten. Anders dagegen ist die Situation in Bild a), wo σ_{prob} zu großzügig mit den teuren Einheiten umgeht. Obwohl σ_{prob} am häufigsten die richtige Diagnose stellt, sind die dabei entstehenden Kosten viel größer als die von σ_{Bay} oder σ_{det}.

5.3 Iterative Bayes-Diagnose

In Abschnitt 5.1 wurde gezeigt, daß mit der Festlegung bestimmter Parameter die probabilistische Diagnose 5.1c in t-diagnostizierbaren Graphen mit Reparatur für alle Syndrome aus $\Omega_1(t)$ gleich und im allgemeinen sogar immer besser oder gleich als die deterministische Diagnose 5.1a diagnostistiziert. Da außerdem ihr Definitionsbereich initial alle Syndrome umfaßt, soll im Folgenden der probabilistischen Diagnose in der allgemeinen Formulierung der Bayesdiagnose der Vorzug gegeben werden.
Ein Problem stellt die für die probabilistische Diagnose und die Bayes- Diagnose nötige Kenntnis der Systemparameter ($p_{ij}, q_{ij}, s_{ij}, r_{ij}$ in Tabelle 1 sowie R_i) dar. Beispielsweise beträgt für das Prime-System die Ausfallrate $\lambda=145\cdot10^{-6}$ pro Stunde /BLO2/. Diese Daten sind meist nicht einfach zu gewinnen. Ein Ausweg bietet die Möglichkeit, das System 'selbstlernend' aus der Zahl der Ausfälle die Systemparameter zu erschließen /BRA1/. Die Tatsache, daß das System nach der Diagnose und Reparatur noch defekt sein kann, kompliziert die Schätzung der Parameter. Trotzdem ergibt sich eine gute Näherung /DIL/.
Dabei bleibt aber ein Problem ungelöst: eine Garantie für eine richtige Diagnose ist nicht gegeben. Diese Tatsache ist bedeutend, wenn mehr als t Fehler in t-diagnostizierbaren Systemen auftreten und auch die probabilistische Diagnose nur eine Diagnostizierbarkeit kleiner eins erreichen kann.
Zur Lösung dieses Problems soll ein sequentielles Diagnoseverfahren mit Reparatur im folgenden Abschnitt beschrieben werden. Das Verfahren benutzt das Prinzip, sich nach einer Reparatur auch von deren Wirksamkeit mit einem Test zu

überzeugen. Dazu betrachten wir zunächst die Bedingungen, unter denen ein Systemtest erkennt, ob das Gesamtsystem fehlerfrei ist.

Sei T_0:= { alle Tests, die 'defekt' als Ergebnis haben }

Sei ein gerichteter Testgraph als 'streng zusammenhängend' bezeichnet, wenn es zwischen je zwei Einheiten u_n und u_m des Graphen immer einen Weg (s./HAR/), sowohl von u_n nach u_m, als auch umgekehrt, gibt. Ein wichtiger Spezialfall sind alle Systeme, in denen Tests zwischen zwei Einheiten in beiden Richtungen möglich sind, z.B. Systeme, deren physikalischen Verbindungen ('Links') bidirektional sind.

Es gilt folgende Feststellung:

FESTSTELLUNG 5.3a

Sei der Testgraph eines Systems streng zusammenhängend und alle Testergebnisse von intakten Einheiten seien korrekt.
Dann ist das System fehlerfrei d.u.n.d, wenn nach einem Systemtest $T_0=\emptyset$ und mindestens eine Einheit korrekt ist.

BEWEIS:

a) Sei das System fehlerfrei. Dann gibt es nach Voraussetzung keine Einheit, die $t_{ij}=1$ hat und damit ist $T_0=\emptyset$.
b) Sei nach einem Systemtest $T_0=\emptyset$ und eine Einheit intakt. Jeder Nachbar, den diese Einheit testet, ist wegen $T_0=\emptyset$ ebenso intakt, ebenso deren Nachbarn und so fort. Da alle Einheiten streng zusammenhängen, gibt es zu jeder Einheit einen Weg von der intakten Einheit, dessen Testergebnisse 'intakt' lauten. Nach Voraussetzung sind damit alle Einheiten intakt. Q.E.D.

Wenn in einem Rechnerverbund ein Rechner, sei er defekt oder nicht, einen Fehler bei einem anderen Rechner (fehlerhaftes Datenübertragunsprotokoll, nicht plausible Datenwerte) oder bei sich selbst (Fehlermeldung im Betriebssystem) findet, so kann dieser Rechner einen Systemtest und Diagnose auslösen (s. Fehlermodell in Kapitel 8.1). Dann ist

T_0= {die Diagnose auslösender Test} U {alle Tests mit $t_{ij}=1$}

und es gilt obiges Lemma sogar ohne die Voraussetzung einer intakten Einheit,

da der vom Initialtest entdeckte Fehler für $T_0=\emptyset$ repariert sein und damit auch mindestens eine intakte Einheit vorliegen muß.

DEFINITION:

Eine Diagnose wird 'überprüft' genannt, wenn folgende algorithmischen Schritte durchgeführt werden:

a) Test
b) Diagnose
c) Systemveränderung (Reparatur,Rekonfiguration)
d) Test, Bildung von T_0

Damit folgt unmittelbar aus Feststellung 5.3a:

FESTSTELLUNG 5.3b:

Sei der Testgraph streng zusammenhängend und alle Testergebnisse von intakten Einheiten seien korrekt.
Dann ist das System nach einer überprüften Diagnose fehlerfrei d.u.n.d., wenn $T_0=\emptyset$ und mindestens eine Einheit intakt ist.

Falls nach einer überprüften Diagnose $T_0\neq\emptyset$ ist, so wird sinnvollerweise nochmals diagnostiziert und damit zu einer 'iterativen Diagnose' übergegangen:

DEFINITION:

Eine iterative Diagnose auf einem System sei durch folgenden Algorithmus definiert:

a) Führe die Tests des Testgraphen durch und bilde T_0.
   Falls $T_0=\emptyset$, STOP.
b) Diagnostiziere auf eine Fehlerklasse
c) Repariere (rekonfiguriere) nach der Fehlerklasse und gehe nach a).

Wie leicht zu sehen ist, stellt die iterative Diagnose eine Erweiterung der überprüften Diagnose dar und garantiert deshalb mit den gleichen Voraussetzungen ein fehlerfreies System, falls der Algorithmus beendet. Wann ist dies nun der Fall ? Unter welchen Voraussetzungen ist die Reparatur erfolgreich mit $T_0=\emptyset$?
Ein spezielles Beispiel für die iterative Diagnose ist die 'sequentielle Diagnose' oder 'Diagnose mit Reparatur' aus Abschnitt 2.1. Sie garantiert

unter der Annahme, daß nicht mehr als t Einheiten defekt sind, in t-diagnostizierbaren Testgraphen mit Reparatur die Tatsache, daß mindestens eine defekte Einheit eindeutig zu lokalisieren ist und mithin keine intakte Einheit irrtümlich ersetzt wird.
Die allgemeine iterative Diagnose stellt geringere Anforderungen an den Testgraphen (s. Feststellung 5.3b). Unter welchen Voraussetzungen hält der Algorithmus nach endlich vielen Schritten an ?
Nehmen wir an, daß in der Diagnose- und Reparaturzeit keine reparierte Einheit ausfällt. Eine hinreichende Voraussetzung ist sicher die Forderung, daß keine Einheit zweimal ersetzt werden soll. Damit läßt sich folgender Algorithmus formulieren:

ITER1:

a) Bilde F:={alle Einheiten}
b) Führe die Tests aus und bilde T_0. Falls $T_0=\emptyset$, STOP.
c) Diagnostiziere auf eine Fehlerklasse F_k unter den Nebenbedingungen $F_k \subset F$ und $F_k \neq \emptyset$, z.B. mit der Bayesdiagnose.
d) Repariere nach F_k und setze $F:=F-F_k$.
e) gehe nach b)

Dieser Algorithmus stoppt nach maximal N Iterationen, da bei jeder Iteration wird mindestens eine Einheit ersetzt wird und die Zahl der möglichen defekten Einheiten in Schritt d) jeweils um mindestens eine Einheit eingeschränkt wird. Bei N Einheiten nach maximal N Reparaturen und damit nach N Iterationen ist das gesamte System ersetzt und damit fehlerfrei.

Die Reparatur von Einheiten kann aus einfachem Austauschen der Einheiten (Austausch von Platinen, Modulen oder gesamten Computeranlagen) bestehen. Ebenso ist aber auch ein Mechanismus denkbar, bei dem defekte Einheiten ab- und intakte spare-Einheiten dazugeschaltet werden (s. Kapitel 9).
Zu beachten ist, daß bei ITER1 außer der Beachtung der angegebenen Nebenbedingungen keinerlei Angaben über den verwendeten Diagnosealgorithmus gemacht wurde. Angenommen, der Diagnosealgorithmus hat nur eine lokale Sicht des Systems (Nachbarschaftstafel!), so kann man nicht a-priori annehmen, daß ein zu allen Tests im System kompatible, systemglobale Fehlerklasse sofort gefunden wird. Besteht zusätzlich die Reparatur bzw. Rekonfiguration des Systems aus dem Ignorieren der defekten Einheiten durch ihre intakten Nachbarn, wie dies für VLSI-Computernetze vorgeschlagen wurde /SNY/ /KOR/, so kann das Netz nach jedem mißlungenen Rekonfigurationsversuch durch Hinzunahme aller 'defekten' Einheiten wieder in den defekten Grundzustand gebracht werden und

ein weiterer Rekonfigurationsversuch unternommen werden. Hierfür eignet sich ITER1 mit der Definition

F:={alle Fehlerklassen}

wobei die Relation $F_k \varepsilon F$ an die Stelle von $F_k \subset F$ tritt. Da es 2^N-1 Fehlerklassen (ausser $\emptyset$) gibt, ist der entsprechende Algorithmus in dieser Version nach maximal 2^N-1 Iterationen beendet. Der Algorithmus garantiert hier allerdings nur ein System, dessen Einheiten entweder alle intakt oder alle defekt sind. Der sich ergebende Endzustand hängt vom verwendeten Diagnosealgorithmus und der vorliegenden Fehlersituation ab.

Das folgende Beispiel soll den Ablauf der iterativen Diagnose bei ITER1 demonstrieren. Sei ein Testgraph nach Abb. 5.3a gegeben mit dem Fehlerzustand $F=\{u_2,u_3\}$.

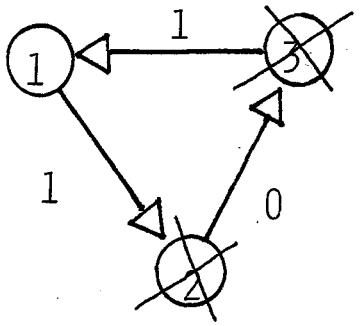


Abb 5.3a Beispiel für ITER1

1.Schritt: $F(1)=\{u_1,u_2,u_3\}$

1.Diagnose: $F_k=\{u_1\}$

$F(2):=\{u_2,u_3\}$

Die erste Diagnose entscheidet falsch. Eine 1-Schritt Diagnose würde nur u_1 ersetzen und damit den Fehlerzustand, das Syndrom und auch eine eventuell nachfolgende 1-Schritt Diagnose unverändert lassen. Die Strategie von ITER1 vermeidet dies:

2.Schritt: $F(2)=\{u_2,u_3\}$

2.Diagnose: $F_k=\{u_2,u_3\}$

$F(3):=\{\}$

Damit ist zwar im ungünstigsten Fall das gesamte System ersetzt, aber erfolgreich repariert worden.

Werden noch die bei der Diagnoseprozedur anfallenden Kosten berücksichtigt (s. Abschnitt 5.2), so kann man als Diagnosevorschrift die Bayes-Diagnose verwenden.
Der Algorithmus lautet damit

ITER2:

a) Bilde $F(1):=\{$alle Einheiten$\}$, $n:=1$.
b) Führe die Tests aus und bilde T_0. Falls $T_0=\emptyset$, STOP.
c) Diagnostiziere auf eine Fehlerklasse F_σ unter den Nebenbedingungen $F_\sigma \subset F(n)$ und $F_\sigma \neq 0$:
Suche F_σ mit $r_\sigma(S)=\min_k r_k(S)$
d) Repariere nach F_σ und setze $n:=n+1$, $F(n):=F(n-1)-F_\sigma$.
e) gehe nach b)

Es ergibt sich somit folgender Sachverhalt:
FOLGERUNG 5.3c:

Sei der Testgraph eines Systems mit N Einheiten streng zusammenhängend und alle Testergebnisse von intakten Einheiten seien korrekt. Dann ist das System mit ITER2 nach maximal N Iterationen bei jedem Syndrom fehlerfrei d.u.n.d., wenn vor der Anwendung des Algorithmus mindestens eine Einheit intakt war.

BEWEIS:

Sei anfangs eine Einheit intakt. Da ITER2 eine Konkretisierung von ITER1 ist, terminiert ITER2 nach maximal N Iterationen mit $T_0=\emptyset$. Dann ist nach Feststellung 5.3a das System intakt.
Sei andererseits keine Einheit intakt. Dann kann das Syndrom mit $T_0=\emptyset$ vorliegen, so daß ITER2 ohne Reparatur terminiert und damit das System defekt läßt. Q.E.D.

6.0 Zentrale und Dezentrale Diagnose

In den vorigen Abschnitten wurden verschiedene Diagnoseverfahren behandelt ohne dabei zu spezifizieren, wer diese Diagnose stellen soll. Die Klärung dieses Problems ist aber sehr wichtig für eine Anwendung der Diagnoseverfahren in Rechenanlagen. Wie in Kapitel 1 näher erläutert wurde, ist es nicht ratsam, diese wichtige Aufgabe einem einzigen Rechner zuzuweisen, da auch dieser defekt werden kann.
Im Gegensatz dazu bieten Systeme von gekoppelten, unabhängigen Rechnern die Möglichkeit, zusätzlich zu dem Test von Rechner zu Rechner ('dezentrales Testen') auch die Diagnose dezentral durchzuführen. Damit ist eine besonders ausfallsicheren Einheit nicht mehr nötig und es wird möglich, durch Beschränkung des Testens und Reparierens auf Teile des Systems (Computerverbunds), den Rechnerbetrieb bei reduzierter Gesamtsystemleistung aufrecht zu erhalten ('Graceful Degradation').
Die dezentralen Diagnoseverfahren ergeben sich nicht sofort aus dem zentralistischen Modell, da viele Verfahren schwer auf dezentrale, asynchrone Entscheidungen übertragen werden können /TEN/. Vielmehr muß bei einzelnen, autonomen Einheiten der gesamte Ansatz neu formuliert werden. Im folgenden Abschnitt sollen nun einige Ansätze zur dezentralen Diagnose geschildert werden.

6.1 Dezentrale Diagnose

Seien N unabhängige Einheiten gegeben, die einen Selbsttest ausführen. Alle Einheiten, die bereits zu diesem Zeitpunkt einen Defekt feststellen, sollen als defekt angenommen und vom Testgraphen sowie den weiteren Überlegungen ausgenommen sein.
Folgende Voraussetzungen sollen gelten:

(6a) Die Testergebnisse von intakten Einheiten seien korrekt (vgl.2.1a), die von defekten nicht vertrauenswürdig.
(6b) Die Kommunikationswege sind intakt.
(6c) Die aufgetretenen Fehler sind dauerhaft.
(6d) Es treten keine neuen Fehler während des Testens auf.

Meyer und Masson konstruierten einen dezentralen Algorithmus /MEY/, der die richtige Diagnose für D_{1t}-Testgraphen findet, falls nicht mehr als t Einheiten defekt sind. Dazu bildet jede Einheit eine Systemtafel (Fehlervektor B_i), in der die Zustände aller anderen Einheiten verzeichnet sind, mit folgendem Algorithmus:

Seien N Einheiten im System und ein D_{1t}-Testgraph mit t vorgegeben.

SELFO: Für jede Einheit u_i gilt:

```
FOR m:=1 TO N DO B_i(m):=0;   (* initialisieren der Systemtafel *)
j:=i;
k:=(i+1) mod N                (* zu testender Nachbar *)
N_F:=0;                       (* Zahl der festgestellten Defekte *)

WHILE N_F<t AND k≠i DO
   IF t_jk=1
      THEN B_i(k):=1; N_F:=N_F+1;
      ELSE j:=k;
   k:=(k+1) mod N;
END_WHILE
```

Die Autoren beweisen, daß mit SELFO jede intakte Einheit die richtige

Systemtafel ermittelt und daß es nur eine Menge der Kardinalität ≥N-t bei N≧2t+1 Einheiten gibt, die identische Systemtafeln haben. Die Diagnose wird dann folgendermaßen durchgeführt:
Die Menge aller Fehlervektoren (B_i), die Matrix B, enthält in jeder Spalte die Urteile (Testergebnisse) aller anderen Einheiten über eine Einheit. Da immer mehr als N-t-1 Einheiten intakt und damit mehr als t Fehlervektoren identisch sind, sind alle Einheiten mit einer Spaltensumme größer als t defekt. Im Gegensatz zu dieser für eine zentrale Auswertungsinstanz formulierten Diagnose entwarfen Kuhl und Reddy /KUH1/ einen dezentralen Algorithmus, in dem Test, Diagnose und Rekonfiguration vollständig dezentral und damit unempfindlich gegenüber dem Ausfall einer beliebigen Einheit ablaufen.

Voraussetzungen sind wieder 6a-6d.

DEFINITION 6.1:

Ein System ist 't-Fehler selbstdiagnostizierbar' genau dann, wenn bei ≦t Fehlern jede fehlerfreie Einheit den Zustand aller anderen Einheiten feststellen kann.

Das Testen im System soll in 'Testrunden' ablaufen; eine Testrunde entspricht einem Test des gesamten Systems und dem Austausch von Nachrichten darüber. Über den zugrunde liegenden Testgraphen wird zunächst keine Aussage gemacht. Angenommen, alle Einheiten haben ihre Nachbarn zu Anfang einer Testrunde getestet. Dann lautet der Diagnosealgorithmus:
Für jede Einheit u_i gilt:

SELF1:

1a) Teste alle Nachbarn.

Bilde eine Systemtafel (Fehlervektor B_i);
nicht getestete Einheiten erhalten '*'.

b) Sende diese Information an alle Testnachbarn.

2) Wenn eine Nachricht von einem Nachbarn kommt:
IF (Nachbar intakt) AND (Einheit mit '*' spezifiziert)
AND (≦t Defekte erkannt)
THEN
a) fülle die Systemtafel mittels der empfangenen Nachricht auf
b) gebe die Nachricht weiter an alle Testnachbarn

3) IF (alle Einheiten spezifiziert) OR (≧t Defekte erkannt)
THEN ersetze alle '*' der Systemtafel durch 'intakt'.

Die Rekonfiguration nach Ablauf von SELF1 wird derart durchgeführt, daß alle defekten Einheiten von weiteren Nachrichten ausgeklammert werden, also kommunikationsmäßig isoliert werden.
Für obigen Algorithmus bewiesen Kuhl und Reddy folgenden Satz:

SATZ 6.1:

Ein System mit dem streng zusammenhängenden Testgrapen G, das SELF1 anwendet, ist genau dann t-Fehler selbstdiagnostizierbar, wenn der Knotenzusammenhang von G (die kleinste Zahl der Knoten, deren Herauslösen den Graphen 'zerfallen' läßt), größer oder gleich t ist.

Der Beweis hierfür benutzt im Wesentlichen die Tatsache, daß nur dann alle intakten Einheiten die gleiche, vollständige Systemtafel erhalten können, wenn es einen Kommunikationsweg gibt, der alle intakten Einheiten miteinander verbindet. Da intakte Einheiten nur Informationen von intakten Einheiten übernehmen, sind mit Voraussetzung 6a ebenso wie im Algorithmus SELF0 die Systemtafeln aller intakten Einheiten identisch und, durch den Zusammenhang, auch vollständig.
Satz 6.1 zeigt im Vergleich zu Satz 2.1 von Hakimi und Amin (Kapitel 2), daß auch für die dezentrale Diagnose mit dem Maximum t an defekten Einheiten der Zusammenhang des Testgraphen eine obere Schranke bildet (vgl. auch Satz 3 in /MAL2/).
Wenn nun Test- und Diagnosezeiten nicht mehr vernachlässigt werden können, gilt Voraussetzung 6d nicht mehr und es können auch während der Ausführung des Algorithmus Defekte aufteten, die die Diagnose fehlerhaft werden lassen. Um dieses Problem zu vermeiden, schlagen die Autoren in der selben Publikation eine modifizierte Version von SELF1 vor:

Für jede Einheit u_i gilt :

Anfangs werden alle Einheiten in der Systemtafel als 'intakt' eingetragen.

$SELF_2$:

1) Teste alle Nachbarn.

IF (Nachbar u_j defekt) THEN
sende an alle anderen intakten Nachbarn diese Nachricht.

2) Wenn eine Nachricht von Nachbar u_j über eine Einheit u_k kommt:
IF (u_j intakt) AND (u_k intakt)
THEN teste u_j;
IF (u_j defekt)
THEN ändere die Systemtafel;
sende 'u_j defekt' an alle intakten Nachbarn
ELSE übernehme die Nachricht über u_k in die Tafel
und gib sie an alle Nachbarn weiter.

Die wichtige Änderung in SELF2 im Vergleich zu SELF1 ist die Vorsichtsmaßnahme, den Nachbarn zu testen, bevor eine Nachricht als wahrheitsgemäß übernommen wird. So kann keine fehlerhafte Diagnoseinformation einer defekten Einheit zirkulieren, egal, zu welchem Zeitpunkt sie defekt wird.
Leider verlangt dieser Algorithmus eine große Anzahl von Tests. Die Autoren interpretieren deshalb das Ausführen eines Tests als eine nur kurze Zeit in Anspruch nehmende Übermittlung der Ergebnisse einer mit Überwachungsaufgaben betrauten, speziellen Hardware bei jeder Einheit.

Um dies zu umgehen, untersuchten die Autoren in einer weiteren Arbeit /KUH2/ Systeme mit Testgraphen, deren Kanten zusätzlich zum Testergebnis noch mit dem Zeitpunkt des Testens gewichtet werden. Die Zeitgewichte garantieren dabei die Korrektheit der Testergebnisse zu einem bestimmten Zeitpunkt. Ein zeitgewichtetes System wird dann '1-Schritt t_a-diagnostizierbar' genannt, wenn jedes Syndrom, das am Ende der Testrunde vorliegt, bei $\leq t_a$ ausgefallenen Einheiten als Ursache nur EINE Fehlerklasse haben kann. Diese Fehlerklasse muß eine Teilmenge der tatsächlich vorliegenden Fehlerklasse sein. In einem solchen System ist die Diagnose also immer richtig, aber nicht unbedingt auch vollständig. Die Autoren geben hinreichende und notwendige Bedingungen dafür an, bei welcher Testreihenfolge (Relation der Zeitgewichte) ein System 1-Schritt t_a- diagnostizierbar ist.

Bei der Untersuchung, unter welchen Bedingungen ein zeitgewichteter Testgraph mit den Annahmen 2.1a,b oder 2.2a 1-Schritt t_a-diagnostizierbar ist, kommen die Autoren zu dem Schluß, daß u.a. jeder D_{1t}-Testgraph (siehe Kapitel 2.1) mit nur einem zusätzlich wiederholtem Test auch zeitgewichtet 1-Schritt t_a-diagnostizierbar ist mit t_a=t.

In der Arbeit /KUH3/ wird von den Autoren der Algorithmus SELF2 für den Fall erweitert, daß zusätzlich zu den Einheiten auch die Kommunikationsverbindungen defekt sein können. Da für das Senden einer Nachricht i.A.

Handshake-Protokolle verwendet werden, ist es sinnvoll, nur Fehler anzunehmen, die auf beide Kommunikationsrichtungen wirken (Leitungsunterbrechung etc.). Zusätzlich wird angenommen, daß die über defekte Verbindungen gesendeten Nachrichten in unbekannter Weise und unbemerkt beeinträchtigt werden. Für das um Verbindungsfehler erweiterte Fehlermodell wird t als $t_{n,1}$ gekennzeichnet. Mit der Definition

> 'Ein System ist genau dann $t_{n,1}$-Fehler selbstdiagnostizierbar, wenn jede fehlerfreie Einheit den Zustand aller anderen Einheiten und den der Verbindungen zwischen fehlerfreien Einheiten feststellen kann; vorausgesetzt, nicht mehr als $t_{n,1}$ Ausfälle (Knoten oder Verbindungen) treten auf.'

formulieren die Autoren in /KUH4/ einen Satz, mit dem dies für ein System nachgeprüft werden kann. Für die Diagnose wird der hinreichende Algorithmus SELF3 angegeben, der hier in Pascal-ähnlicher Syntax widergegeben werden soll. Für jede Einheit u_i gilt:

SELF 3:

Initialisiere die Systemtafel (Einheiten und Verbindungen link(u_1,u_k)) mit 'alles intakt'.

Teste alle Nachbarn.

1) Falls Fehler entdeckt wurden, teile dies allen anderen als intakt angesehenen Nachbarn, zu denen 'intakte' Verbindungen bestehen, mit.

2) Eine Nachricht trifft bei u_i ein mit dem Inhalt: 'u_1 hält u_k für defekt'.

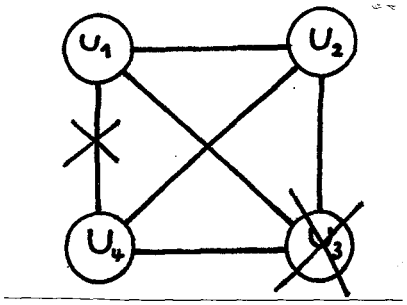
```
IF      (Absender u_j ist Nachbar von u_i)
    AND (u_j ist als 'intakt' registriert)
    AND (u_1 ist Nachbar von u_k)
    AND (Der Inhalt ist neu )
  THEN
      Teste den Nachbarn u_j. Falls defekt, GOTO 1).
      IF (u_k ist nicht Nachbar von u_i)
         THEN übernehme die Nachricht und GOTO 1).
         ELSE
```

Teste Nachbar u_k
IF (u_k intakt)
THEN link(u_1,u_k):=defekt, GOTO 1).
ELSE übernehme die Nachricht, GOTO 1).

3) Eine Nachricht trifft bei u_i ein mit dem Inhalt:
'link(u_1,u_k) ist defekt'.

IF (Absender u_j ist Nachbar von u_i)
AND (u_j ist als 'intakt' registriert)
AND (Der Inhalt ist neu)
THEN
Teste den Nachbarn u_j. Falls defekt, GOTO 1).
Registriere die Nachricht.
IF ('u_k hält u_1 für defekt' ist registriert)
THEN lösche diese Information, GOTO 1).

Dieser Algorithmus kann bei dauerhaften Fehlern an Knoten und Verbindungen angewendet werden. Durch die erneuten Tests des Nachbarn nach Erhalt einer Nachricht werden auch die bei einer Testrunde neu auftretenden Fehler richtig erkannt (vgl.SELF2).
Durch den dritten, speziell für Verbindungsfehler entwickelten Teil testen bei einem vermuteten Defekt an einer Einheit alle direkten Nachbarn diese ebenfalls. Entstehende Differenzen zwischen den Testern werden nicht, wie bisher in den graphentheoretischen Modellen, als Zeichen für den Defekt aller Tester, sondern für einen Defekt der Verbindungen unter ihnen gewertet.
Bei transienten Fehlern versagt SELF3, da jeder transient defekter Knoten als Dauerdefekt der entsprechenden Verbindung gewertet wird.
Beispiel für SELF3:



Defektes System aus 4 Einheiten

Sei u_3 und die Verbindung zwischen u_1 und u_4 defekt,
Sei $t_{ij}=\begin{cases} 0 & u_j \text{ ist intakt'} \\ 1 & u_j \text{ ist defekt'} \end{cases}$

$t_i:=(t_{ij})_{j=1..N}$
und F die diagnostizierte Fehlerklasse.
Nach Schritt 1 ist bei
Einheit 2: $t_2=(0,0,1,0)$, $F=\{u_3\}$,
Einheit 1: $t_1=(0,0,1,1)$, $F=\{u_3,u_4\}$
Einheit 4: $t_4=(1,0,1,0)$, $F=\{u_3,u_1\}$

Betrachten wir Einheit 2. Zuerst teilt u_2 seine Testergebnisse den als 'intakt' angesehenen Nachbarn u_1 und u_4 seine Testergebnisse mit. Dann übernimmt es die Testergebnisse von u_1 und u_4 und gibt sie weiter. Die Ergebnisse von u_3 werden ignoriert, da sie als 'defekt' angesehen werden. Da sowohl u_1 als auch u_4 als 'intakt' registriert sind, wird die Kommunikationsverbindung $link(u_1,u_4)$ und $link(u_4,u_1)$ als 'defekt' angesehen, dies registriert und u_1 und u_4 mitgeteilt.
Betrachten wir nun u_1 und u_4. Diese Einheiten empfangen zwar beide die Testergebnisse t_{2j}, bemerken aber nicht die Differenz zu ihrer eigenen Fehlerklasse F. Erst mit der Nachricht von u_2 'link(u_1,u_4)=defekt ' und link(u_1,u_4)=defekt' wird die Fehlerdiagnose geändert, so daß schließlich bei allen drei intakten Einheiten die einheitliche Diagnose resultiert

Einheit 1: $F=\{u_3, link(u_4,u_1), link(u_1,u_4)\}$
Einheit 2: $F=\{u_3, link(u_4,u_1), link(u_1,u_4)\}$
Einheit 3: $F=\{u_3, link(u_4,u_1), link(u_1,u_4)\}$

In /MAEH/ wird eine weiterentwickelte Version von SELF1 beschrieben, die ohne eine zusätzliche Kommunikation wie SELF3 auch Verbindungsfehler diagnostizieren kann. Dazu überprüft jede Einheit u_k die Matrix der Testergebnisse (t_{ij}) darauf, welche der als 'intakt' registrierten Einheiten sich gegenseitig für 'defekt' halten. Die Abbruchbedingung für eine Testrunde wird so festgelegt, daß von jeder als 'intakt' registrierten Einheit im System die Testergebnisse über alle Nachbarn eingetroffen sein müssen. Ist dies erreicht, so werden die Testergebnisse diagnostiziert.
Weitere Algorithmen finden sich bei /PRA/, /KUH4/.

6.2 Dezentrale, iterative Bayesdiagnose

In dem vorigen Abschnitt wurden einige Grundgedanken zu der Problematik der dezentralen Diagnose erörtert und die Algorithmen SELFO-3 vorgestellt. Im Folgenden soll nun der Versuch unternommen werden, diese Grundgedanken in Hinblick auf eine probabilistische Diagnose mit Kosten (Bayes-Diagnose) zu erweitern.

Bei einer dezentralen Diagnose existiert keine zentrale Instanz, die mit Hilfe des aktuellen Syndroms eine Diagnose stellen kann. Stattdessen muß jede Einheit für sich allein entscheiden, welche Einheiten sie für defekt hält.
Es ist möglich, mit Hilfe eines Konsens-Mechanismus /ROB/ diese privaten Entscheidungen zu verallgemeinern, doch ist dies nur bei stark gekoppelten Einheiten sinnvoll /TEN/, da sonst die Kommunikationskanäle durch die Nachrichten zum Abstimmen aller Einheiten untereinander zu stark belastet werden.
Ein anderer Weg besteht darin, jede Einheit die Meldungen anderer Einheiten über gefundene Fehler geeignet interpretieren zu lassen. In den vorher erörterten Arbeiten geschieht das dadurch, daß mit der Entscheidung, nur Nachrichten von selbst getesteten und für intakt befundenen Einheiten zu vertrauen, innerhalb des Testgraphen mit der Nachrichtenkette ein Teilgraph von intakten Einheiten gebildet wird.
Bei der probabilistischen Diagnose ist das in dieser Form nicht mehr möglich, da i.A. auch eine intakte Einheit nicht alle Fehler einer defekten Einheit findet und damit keinen absolut vertrauenswürdigen Nachbarn hat, über den sich ein Konsens bilden läßt. Einen Lösungsweg für dieses Problem zeigt der Ansatz von Pease, Shostak und Lamport /PEA1/(s.a./PEA2/,/DOL/). Die Autoren zeigen, daß es bei N Einheiten in einem vollständig vernetzten Graphen möglich ist, einen Konsens herzustellen, wenn nicht mehr als $m \leq (N-1)/3$ Einheiten defekt sind. Dabei muß allerdings die Information in m+1 Runden ausgetauscht werden. Eine Alternative dazu bietet die Codierung der Information mit kryptographischen Methoden.Hierbei muß nur noch $N \geq m \geq 0$ gelten.
In Rechnernetzen müssen Codierungsmöglichkeiten schon für den 'elektronischen Briefverkehr' vorhanden sein, um die Authentizität des Absenders zu gewährleisten /DIF1/,/DIF2/. Dies ist in einem 'Public-key Cryptosystem' gegeben, bei dem zwar der Entschlüsselungsalgorithmus E und der Schlüssel jedes Absenders bekannt sind ('public'), aber nicht der Verschlüsselungsmechanismus V einer Nachricht M. Ein solches Kryptosystem wird dabei mit folgenden Forderungen festgelegt:

1) E(V(M)) = M
2) E und V sollen einfach zu berechnen sein
3) Wenn V bekannt ist, so ist damit keine einfache Methode gegeben, E zu berechnen.
4) V(E(M)) = M

In ihrer klassischen Arbeit /RIV/ gaben als Erste Rivest, Shamir und Adleman Algorithmen an, die obige Forderungen verwirklichen. Den Begriff 'keine einfache Methode' maßen sie an der schnellsten Methode, die Codierung zu herauszufinden: einer Faktorisierung durch Primzahlen. Mit dem schnellsten bekannten Algorithmus und einem Rechner mit einer Mikrosekunde pro Operation dauert dies bei einem Schlüssel von 200 Stellen immerhin $3.8 \cdot 10^9$ Jahre, also ausreichend lange für den Schutz der meisten Anwendungen.
Bei der betrachteten Anwendung für Testzwecke reicht in den meisten Anwendungsfällen sicherlich die Maßnahme aus, jeder Einheit die eigene Nummer nur in verschlüsselter Form im ROM mitzugeben. Damit kann jeder Empfänger die Legalität des Absenders prüfen, aber ohne Verschlüsselungsalgorithmus keine falsche Nummer produzieren. Die Wahrscheinlichkeit, daß eine defekte Einheit 'intelligent und bösartig' aus einer eingegangenen Nachricht den Identifizierungscode in den entsprechenden Teil einer auszugebenden Nachricht kopiert, kann als vernachlässigbar klein angesehen werden.
Für die Testzwecke reicht folgende Eigenschaft aus:
Sei eine kryptographische Verschlüsselungsmöglichkeit gegeben, so daß jede verschlüsselte Nachricht zwar von jeder Einheit entschlüsselt werden, aber nicht genauso verschlüsselt werden kann. Damit läßt sich eindeutig von der Verschlüsselung auf den Absender der Nachricht schließen.

Betrachten wir nun den Diagnosealgorithmus. Die Diagnose soll wieder in 'Testrunden' ablaufen.

Für jede Einheit u_i gilt:

BAYO:

1) Initialisiere die Systemtafel (alle Einheiten :='intakt')
und die Syndromtafel.
Teste alle Nachbarn.
Verschlüssle die Ergebnisse
und sende sie an alle Nachbarn, die 'intakt' sind.

2) Wenn eine Nachricht eintrifft:
IF (Nachrichtenformat ok) AND (Nachricht ist neu)
THEN
Vervollständige die Syndromtafel.
Sende die Orignalnachricht weiter an alle Nachbarn,
die 'intakt' sind.

3) Wenn die Testrunde zu Ende ist:
Führe eine Bayes-Diagnose durch:
Wähle die Fehlerklasse F_k so, daß für alle anderen F_j gilt
$r_k(S) \leqq r_j(S)$
Ändere die Systemtafel mit F_k(z.B.Reparatur)

Im obigen Algorithmus diagnostiziert jede Einheit alle anderen, wobei die Entscheidung für eine Fehlerklasse nicht zweifelsfrei, sondern mit Hilfe errechneter Wahrscheinlichkeiten und Kosten getroffen wird. Um Aussagen über diesen Algorithmus formulieren zu können, ist es vorher nötig, den Begriff 't-Fehler selbstdiagnostizierbar' mit einer Neudefinition zu erweitern:

DEFINITION 6.2:
Ein System heißt 't-Fehler selbstdiagnostizierbar' genau dann, wenn bei weniger als t Fehlern der Zustand aller Einheiten von jeder intakten Einheit gleich diagnostiziert wird.

Diese Definition stimmt bei der Vollständigkeit der Tests und der Anwendung von beispielsweise der probabilistischen Diagnose mit Definition 6.1 von Kuhl und Reddy überein, da in diesem Fall die Diagnose aller intakten Einheiten korrekt ist. Wird dagegen die Bayesdiagnose verwendet, so wird die kostengünstigste Diagnose errechnet, die nicht unbedingt auch korrekt sein muß (siehe Kapitel 5.2).
Damit läßt sich folgender Satz formulieren:
SATZ 6.2a
Seien mit den Parametern der Einheiten bei beliebigem Syndrom S und Fehlerklasse F die Wahrscheinlichkeit P(S,F) bekannt.
Ein System mit N Einheiten und dem Testgraphen G, das eine Nachrichtencodierung mit obigen Eigenschaften und den Algorithmus BAYO anwendet, ist t-Fehler selbstdiagnostizierbar bei allen Parametern der

Einheiten d.u.n.d., wenn der Zusammenhang von G größer als t ist.

BEWEIS:

Falls überhaupt formattreue Nachrichten eintreffen, stammen sie nach Voraussetzung von den angegebenen Absendern und sind damit authentisch. Da intakte Einheiten zwar nicht notwendigerweise richtige Testergebnisse erzielen, aber Nachrichten weiterleiten können, werden in der durch die intakten Einheiten gebildete Nachrichtenkette alle Testergebnisse an alle angeschlossenen intakten Einheiten weitergeleitet.

a) Notwendigkeit:

Angenommen, der Zusammenhang von G ist $\leq t$.

Dann gibt es bei t Fehlern zwei intakte Einheiten im System, zwischen denen nur Wege existieren, die über defekte Einheiten führen (Satz von Menger, /HAR/,Theorem 5.10) Der Testgraph zerfällt bei t Fehlern somit in mindestens zwei Untergraphen. Da ein Austausch der Testergebnisse nicht mehr möglich ist, sind i.A. (bei $P(t_{ij}=0/e_i=0,e_j=0)\neq 1$, wobei e_i den Zustand von u_i bezeichnet) die Syndrome und damit auch die Diagnosen nicht übereinstimmend; jede Einheit diagnostiziert nur jene Einheiten als defekt, über die entsprechende Testergebnisse vorliegen.

b) hinreichend:

Angenommen, der Zusammenhang von G ist $> t$.

Dann existiert bei $\leq t$ Fehlern immer mindestens ein Weg zwischen je zwei intakten Einheiten, der nur über intakte Einheiten führt und damit einen Austausch der Testergebnisse, Gleichartigkeit der Syndrome und somit bei Gleichartigkeit des Diagnoseverfahrens auch gleiche Diagnosen garantiert.

Q.E.D.

Wie leicht ersichtlich, gilt obiger Beweis nicht nur für BAYO, sondern für alle Diagnosealgorithmen, die bei $P(t_{ij}=0/e_i=0,e_j=0)\neq 1$ definiert sind. Ist dagegen $P(t_{ij}=0/e_i=0,e_j=0)=1$ garantiert und werden alle Einheiten, über die kein Testergebnis vorliegt, als intakt angenommen, so werden bei einem streng zusammenhängenden Testgraphen mit dem Knotenzusammenhang t und t fehlerhaften Einheiten alle defekten Einheiten von jedem Teilgraphen von streng zusammenhängenden, intakten Einheiten getestet und als defekt erkannt. Trotz fehlendem Informationsaustausch ermitteln die intakten Einheiten jedes Teilgraphen das gleiche Syndrom und damit die gleiche Diagnose (s. Satz 1 von Kuhl, Reddy in /KUH1/).

Wenn in einem Testgraphen zusätzlich zu den defekten Knoten (Prozessoreinheiten) auch die Kanten (Kommunikationswege) defekt sein können, gilt obiger Satz auch in der erweiterten Form:

SATZ 6.2b:

Ein System mit dem Testgraphen G, das den Algorithmus BAYO anwendet, ist t-Fehler selbstdiagnostizierbar d.u.n.d., wenn der Zusammenhang von G größer als t ist, wobei t=Zahl der defekten Knoten + Zahl der defekten Kanten.

BEWEIS:

Notwendigkeit:

Sei der Zusammenhang von G $\leq$t und N$\geq$t+2.

Dann gibt es mindestens zwei Einheiten im System, zwischen denen nur maximal t kreuzungsfreie, verschiedene Wege existieren. Bei t Fehlern können alle t Wege durch defekte Knoten ODER defekte Kanten ausfallen, so daß diese beiden Einheiten bei $P(t_{ij}=0/e_i=0,e_j=0)\neq 1$ ungleiche Syndrome und Diagnosen ermitteln können.

Hinreichend:

Es gilt sinngemäß der Beweis von Satz 6.2a.

Nach Theorem 5.1 von Harary /HAR/ ist der Kantenzusammenhang eines Graphen größer oder gleich dem Knotenzusammenhang. Diese potentielle Fehlertoleranz läßt sich aber für Testgraphen im obigen Satz nicht ausnutzen, da bei Mischfehlern die schwächeren Eigenschaften des Knotenzusammenhangs die Fehlertoleranzeigenschaften begrenzen.

Im Unterschied zu den überlegungen von Kuhl und Reddy (/KUH3/,p.101) ist in Satz 6.2b die Fehlersituation von einem defekten Knoten, verbunden mit einer defekten Kante, nicht extra abgehoben. Zweifelsohne kann in diesem Fall der Zustand der Verbindung nicht einwandfrei festgestellt werden. Dies ist aber für die Definition 6.2 von 't-Fehler selbstdiagnostizierbar' nicht unbedingt nötig. Es reicht aus, daß alle intakten Einheiten die gleiche Diagnose stellen, was mit Satz 6.2b auch gegeben ist.

Um den Reparaturerfolg zu überwachen, ist es auch sinnvoll, die Bayesdiagnose iterativ durchzuführen. Dazu wird die dezentrale Version von ITER2 gebildet:

Sei T_0 die Menge aller Tests mit dem Ergebnis 'defekt'(s.Kapitel 5.3).

Angenommen, die reparierten Einheiten bleiben während der Diagnose intakt.

Für jede Einheit u_i gelte

BAY1:

Testrunde:=1

Initialisiere die Systemtafel (alle Einheiten :='intakt')

1) Initialisiere die Syndromtafel und T_0.
Teste alle Nachbarn.
Verschlüssle die Ergebnisse mit der Nummer der Einheit und der Testrunde und sende sie an alle Nachbarn, die 'intakt' sind.

2) Wenn eine Nachricht eintrifft:
IF (Nachrichtenformat ok) AND (Nachricht ist neu)
THEN
Vervollständige die Syndromtafel und T_0.
Sende die Orignalnachricht weiter an alle Nachbarn, die 'intakt' sind.

3) Wenn die Testrunde zu Ende ist:
IF (T_0=Ø)
THEN Stop.
ELSE
IF (Testrunde=1) THEN F(1):={alle Einheiten}.
Führe eine Bayes-Diagnose durch:
Wähle die Fehlerklasse F_k cF(Testrunde) so,
daß für alle anderen F_j cF(Testrunde) gilt
$r_k(S) \leq r_j(S)$.
Repariere nach F_k.
Testrunde:=Testrunde+1; F(Testrunde):=F(Testrunde-1)-F_k.
GOTO 1).

Im obigen Algorithmus wird wie bei ITER2 die Zahl der möglicherweise defekten Einheiten stetig verringert, da eine Diagnose ohne Reparatur nur einmal zugelassen wird. Es werden in jeder Testrunde neue Tests durchgeführt, neue Syndrome erschlossen und eine genauere Diagnose durchgeführt. Durch die stetige Verringerung von F(n) terminiert der Algorithmus wie ITER2 in maximal N Schritten.
Da die reparierten Einheiten während der Diagnose intakt sind, muß spätestens bei F(n)=Ø die Abbruchbedingung T_0=Ø vorliegen.

Für obigen Algorithmus gilt folgender Satz:

SATZ 6.2c:

Seien die Voraussetzungen 6a)-d) erfüllt und das System t-Fehler

selbstdiagnostizierbar.

Wenn $\leq t$ Fehler vorliegen, so ist nach Ausführung von BAY1 das System fehlerfrei.

BEWEIS:

In t-Fehler selbstdiagnostizierbaren Systemen ist nach Satz 6.2a mit BAY0 in allen intakten Einheiten die gleiche Systemtafel gegeben. Da BAY1 den Algorithmus BAY0 in iterativer Formulierung enthält, gilt obiges auch für BAY1.

Nach Terminierung von BAY1 ist somit bei allen intakten Einheiten $T_0=\emptyset$ und eine gleiche Systemtafel vorhanden. Somit läßt sich Feststellung 5.3a anwenden; das System ist fehlerfrei Q.E.D.

Die bisher vorgestellten Algorithmen benutzen als grundsätzlichen Rahmenbegriff den Ausdruck 'Testrunde'. Was ist darunter zu verstehen?

Eine Testrunde ist der Zeitabschnitt zwischen Beginn und Ende eines Systemtests mit Diagnose und Reparatur. Dies verlagert das Problem auf die Frage, wie eine Testrunde eingeleitet und wie sie beendet wird.

Über die Einleitung einer Testrunde wird in allen zitierten Publikationen nichts näheres erwähnt; es soll deshalb innerhalb eines Modells im Abschnitt 8 näher betrachtet werden.

Für die Beendigung einer Testrunde lassen sich verschiedene Kriterien angeben /KUH1/. In SELF1 wird die Testrunde beendet, wenn

a) die Systemtafel (bzw. Syndromtafel) vollständig spezifiziert ist
oder b) bereits t Fehler gefunden worden sind.

Bei mehr als t Fehlern ist dies aber nicht immer gewährleistet, da dann eine intakte Einheit informationsmäßig isoliert sein kann; SELF1 wird bei einer solchen Einheit nie terminieren.

Deshalb empfiehlt es sich, zur Beendigung der Testrunde zusätzlich eine Zeitüberwachung einzuführen (s. Abschnitt 8.1). Dies ist auch bei SELF2 nötig, bei dem von seinen Autoren kein Abbruchkriterium angegeben wird; da die Systemtafel auf 0 initialisiert ist, entfällt das wichtigste Kriterium a).

In /MAEH/ wurde eine andere Lösung des Problems vorgestellt. Eine Testrunde gilt dort als beendet, wenn

c) von jeder Einheit, die als 'intakt' registriert ist, auch eine Testaussage über ihre Nachbarn vorliegt.

7.0 Lokale Diagnose und Rechnernetze

In den bisherigen Betrachtungen wurde jeweils die Existenz eines durch einen Test des gesamten Systems erlangten Syndroms vorausgesetzt. Die Tests werden dazu gleichzeitig oder hintereinander ausgeführt, ohne daß dabei Benutzerprogramme laufen, wie es beispielsweise beim Starten eines Systems geschehen kann. Im Folgenden soll die Problematik dieser Voraussetzung diskutiert werden und Alternativen dazu erörtert werden.

7.1 Lokales Testen

Wenn dem Multicomputersystem lebenswichtige Aufgaben übertragen worden sind (s. Kapitel 1), so ist es nicht tragbar, wenn das gesamte System durch Testprogramme gleichzeitig benutzt und damit blockiert wird. Eine andere Möglichkeit besteht darin, die Testprogramme hintereinander auf verschiedenen Einheiten und parallel zum normalen Betrieb im System bearbeiten zu lassen. Die Diagnose wird nach Durchführung des letzten Tests vorgenommen. Dies vermeidet zwar eine Blockierung des Systems, schafft aber neue Probleme, da ein defekter Prozessor vor seiner Reparatur fehlerhafte Daten erstellen und weitergeben kann. Selbst bei einer korrekten Diagnose ist es doch nicht mehr möglich, die falschen Daten und die davon mit korrekten Programmen auf intakten Einheiten gewonnenen falschen Ergebnisse zu korrigieren, da auch diese Ergebnisse bereits weitergegeben sein können. Die einzige Möglichkeit, alle Fehlerauswirkungen zuverlässig zu beseitigen, ist eine Initialisierung des Gesamtnetzes, was aber bei zeitkritschen, lebenswichtigen Systemen sehr problematisch ist.
Eine andere Möglichkeit besteht darin, Fehlererkennungsmechanismen derart in das System einzubauen, daß auftretende Fehler bereits im Normalbetrieb erkannt werden (s.a.Abschnitt 8.1) und im Fehlerfall weitere Tests einzuleiten. Dies ist besonders in Computernetzen mit beschränkter Kommunikation, und damit auch beschränkter Fehlerfortpflanzung, sinnvoll. Es wird nur der Bereich des Systems getestet, in dem der Fehler aufgetreten ist, während die restlichen Einheiten des Systems die wichtigsten Benutzertasks abarbeiten.
Als Beitrag zu diesen Überlegungen sei auf die Arbeit von Saheban, Simoncini und Friedman /SAH/ verwiesen. In ihr wird untersucht, wie die Einheiten eines D_{1L}-Graphen, einer Verallgemeinerung der D_{1A}-Graphen aus Kapitel 2.4 mit L Tests pro Einheit, verbunden sein müssen, um bei möglichst vielen, mit Benutzertasks beschäftigten Einheiten eine t-Fehler Diagnostizierbarkeit im Testgraphen der übrigen Einheiten zu erreichen. Sind die B beschäftigten

Einheiten nicht frei wählbar, so ergibt sich bei N Einheiten und L von jeder Einheit getesteten Nachbarn ein minimales t von t=min(L-B, $\lfloor (N-B-1)/2 \rfloor$). Lassen sich dagegen die Benutzertasks den Einheiten frei zuordnen, so sind bessere Bedingungen ableitbar.
Im folgenden Abschnitt wird eine wichtige Klasse von Rechnernetzen eingeführt, in der die oben genannte Eigenschaft der begrenzten Fehlerfortpflanzung gegeben ist. In Abschnitt 7.4 wird sodann die t-Diagnostizierbarkeit dieser Rechnernetze untersucht.

7.2 Reguläre Computernetze

Für VLSI-Anwendungen ist besonders die Klasse der regulären (d.h. regelmäßigen) Flächennetze geeignet (s. Kapitel 1). Durch die Replikation gleichartiger Rechen- und damit auch Flächenelemente wird kostengünstiges Lay-out, Herstellung, Testen und Software-Erstellung möglich.
Damit sich die Struktur der Flächennetze leicht auf der Ebene eines Siliziumscheibchens abbilden läßt, eignen sich deshalb besonders als Kommunikations- (und damit auch als Teststrukturen) kreuzungsfreie, reguläre Flächennetze. Da alle Rechnereinheiten gleiche Bauart haben sollen, ist die Zahl der Kommunikationsverbindungen pro Einheit ebenfalls gleich groß. In Abb.7.1a ist ein Ausschnitt aus einem solchen kreuzungsfreien Rechnernetz zu sehen.

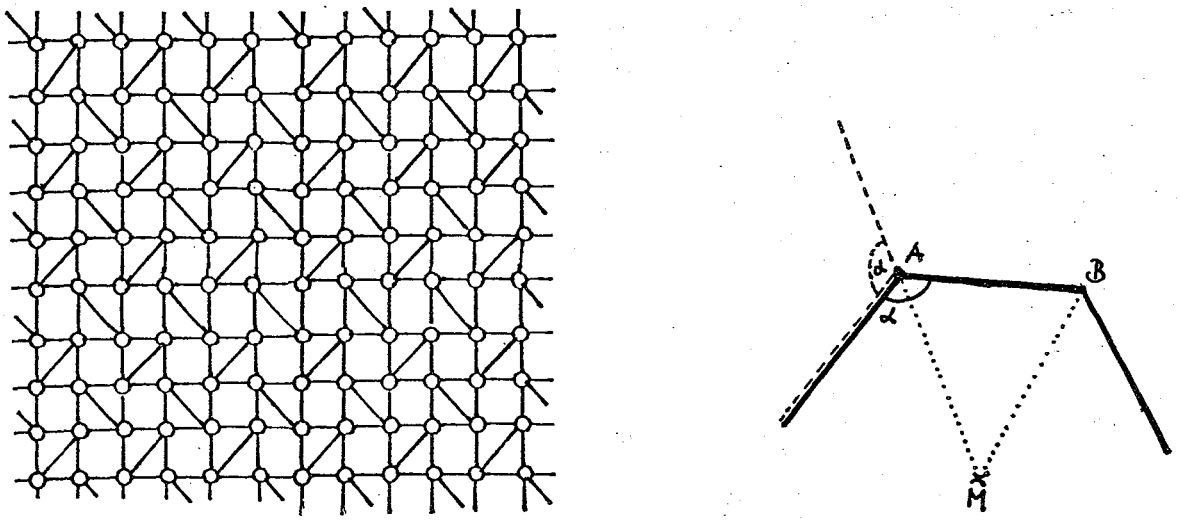
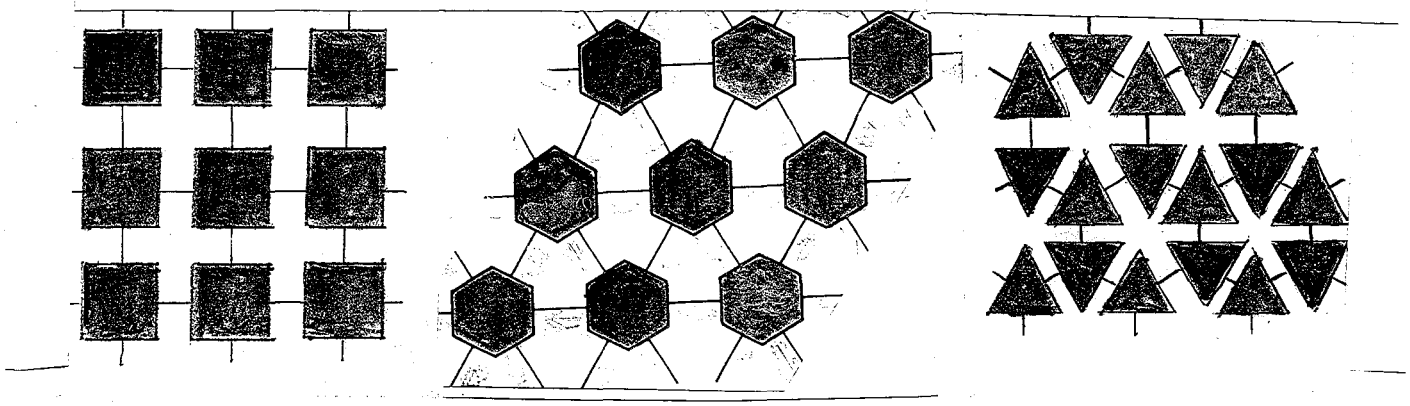


Abb. 7.1 a) Kreuzungsfreies Rechnernetz mit 5 Nachbarn pro Einheit b) Ecke eines Flächenelements

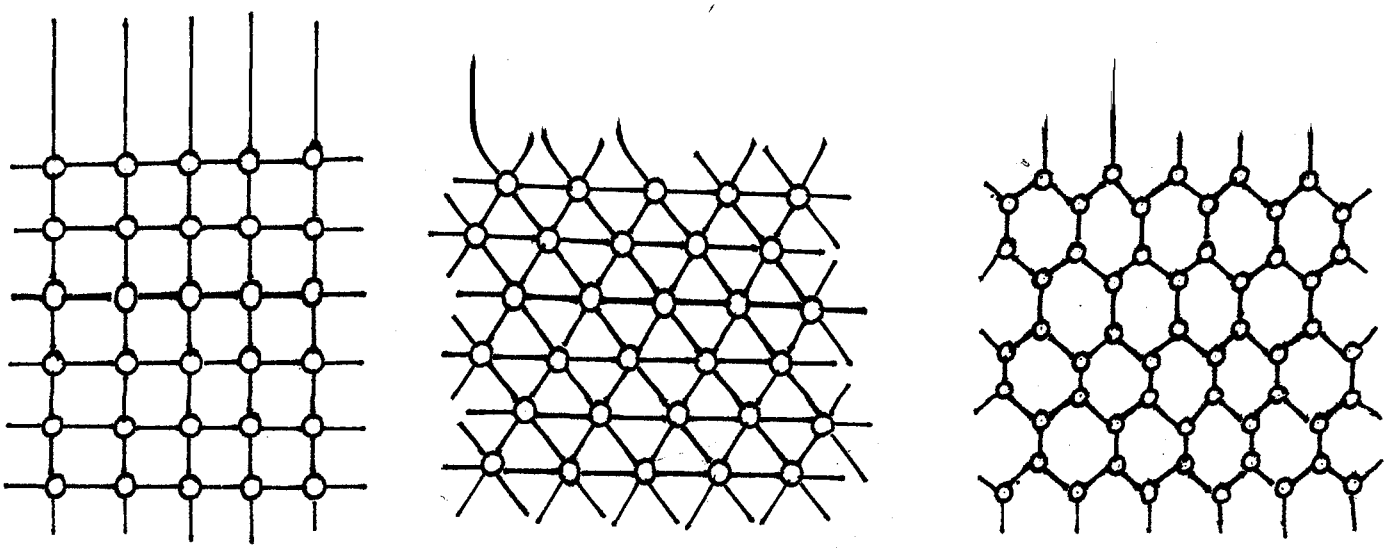
Da für das VLSI-Layout hauptsächlich die Replikation gleicher Flächenelemente

wichtig ist, sei als weitere Nebenbedingung die Forderung gestellt, daß die Flächenelemente regelmäßige Vielecke eines Typs sein sollen.
Betrachten wir nun eine Ecke eines solchen Vielecks im Punkt A (Abb. 7.1b). Sei α der Winkel zwischen den zwei Kanten. Mit welchen n-Ecken ist eine regelmäßige Partition der Ebene möglich?
Da nur n-Ecke gleichen Typs mit gleichem Innenwinkel α verwendet werden und jede Einheit gleiche Zahl von Nachbarn hat, haben jeweils m der n-Ecke einen Eckpunkt gemeinsam und es gilt $m\alpha=360$ Grad. Da die n-Ecke auch gleiche Kantenlängen haben, ist die Summe der Innenwinkel $\alpha+360/n = 180$ für das Dreieck (A,M,B). Also gilt für alle gesuchten n-Ecke $m=360/\alpha=2n/(n-2) =: f(n)$. Da die Funktion $f(n)$ stetig fallend ist und $m\in\mathbb{N}$ $m>1$ gelten muß, erfüllen nur wenige n-Ecke die gestellten Bedingungen. Für n=3 (Dreieck) ist m=6, für n=4 (Viereck) ist m=4, für n=5 (Fünfeck) ist m=3.333 und bei n=6 (Sechseck) ist m=3. Bei m=2 ist n=n-2; es existiert kein n-Eck mit diesen Bedingungen.
Also eigenen sich für VLSI-Anwendungen besonders Flächennetze, deren Rechenelemente die Fläche eines Dreiecks, Vierecks oder Sechsecks besitzen. In Abb. 7.1 c,d,e ist die Flächenaufteilung der VLSI- Flächenelemente zu sehen, während Abb.7.2a,b,c Ausschnitte aus der Verbindungsstruktur der Netze zeigt.



c) d) e)

Abb 7.1c,d,e VLSI-Flächenelemente regulärer Flächennetze



a) b) c)

Abb 7.2a,b,c Verbindungstypen regulärer Flächennetze

Bestimmte VLSI-Netztypen werden auch als 'systolische Felder' bezeichnet /KUN/ und eignen sich hervorragend zur schnellen Matrizenmultiplikation (s.Abb.7.2e). Ein großer Teil der für Signalverarbeitung nötigen Rechnungen kann auf einige, wenige Matrizenoperationen, wie Multiplikation, Inversion, etc zurückgeführt werden /KUNG/. Damit sind die Flächennetze nicht nur für die numerische Datenverarbeitung als Spezialmaschinen einsetzbar, sondern auch für das breite Anwendungsgebiet der Signalverarbeitung.

Der Netztyp a) bietet sich mit seiner Struktur für Matrizenrechnungen an; der bekannteste Vertreter dieses Typs ist unter dem Namen 'Holland-Rechner' bekannt /MIE/.

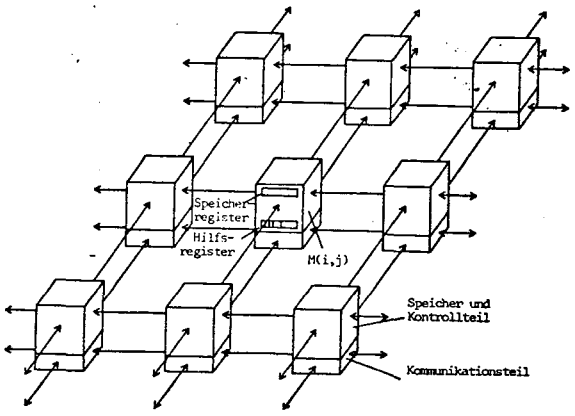


Abb 7.2d Holland-Rechner

Der Netztyp b) läßt sich ebenfalls für schnelle Matrizenmultiplikation benutzen, wie in Abb.7.2e aus /MEA/ dargestellt. Die Argumente werden dazu an zwei Gitterseiten eingespeist; das durch paralleles Rechnen und Pipelining schnell ermittelte Ergebnis erhält man von der dritten Seite.

$$\begin{bmatrix} a_{11} & a_{12} & & & 0 \\ a_{21} & a_{22} & a_{23} & & \\ a_{31} & a_{32} & a_{33} & a_{34} & \\ & a_{42} & \ddots & & \\ 0 & & & & \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & b_{13} & & 0 \\ b_{21} & b_{22} & b_{23} & b_{24} & \\ & b_{32} & b_{33} & b_{34} & b_{35} \\ & & b_{43} & \ddots & \\ 0 & & & & \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} & 0 \\ c_{21} & c_{22} & c_{23} & c_{24} & \\ c_{31} & c_{32} & c_{33} & c_{34} & \\ c_{41} & c_{42} & & \ddots & \\ 0 & & & & \end{bmatrix}$$

A B C

Mit der Iterationsvorschrift für jedes Rechenelement

$c_{ij}(1) := 0;$

$c_{ij}(k+1) := c_{ij}(k) + a_{ik}\, b_{kj}$

$c_{ij} := c_{ij}(n+1)$

werden zwei Matrizen A,B, vorwiegend sog. 'Bandmatrizen', miteinander multipliziert.

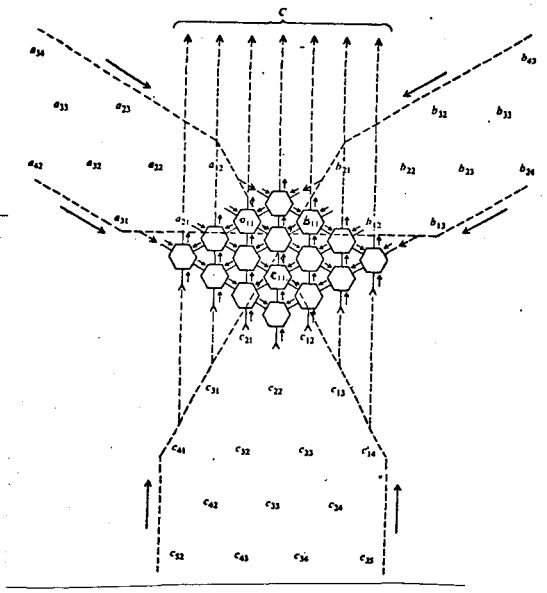


Abb 7.2e Matrizenmultiplikation (aus /MEA/)

Die bisher beschriebenen Rechnernetze dienen speziellen Anwendungen.
Wie kann ein allgemeiner Multiprozessorchip aussehen?
Eine Möglichkeit besteht darin, zusätzlich zu den Recheneinheiten Schalteinheiten vorzusehen /SNY/ (s. Kapitel 1, Abb. 1e,f). Durch die Erweiterung der Prozessorelemente um Schaltelemente, die es gestatten, beliebig Ein- und Ausgänge zum verzögerungsfreien Signaltransport zusammenzuschalten, wird aus dem Netz mit fester Struktur ein beliebig konfigurierbarer, fehlertoleranter Rechner.
Eine weitere Möglichkeit besteht darin, im Unterschied zu /SNY/ die Schaltfunktionen nicht als gesonderte Schalter auszuführen, sondern jedes Rechenelement zusätzlich mit Schaltfunktionen auszustatten. Dadurch wird nur ein Typ von VLSI- Flächennelementen benötigt und die Flächenaufteilung entspricht der Abbildung 7.1c,d,e (vgl. /KOR/).
In Abb. 7.2f ist die Realisierung einer Baumstruktur auf einem solchen Rechnernetzwerk dargestellt.

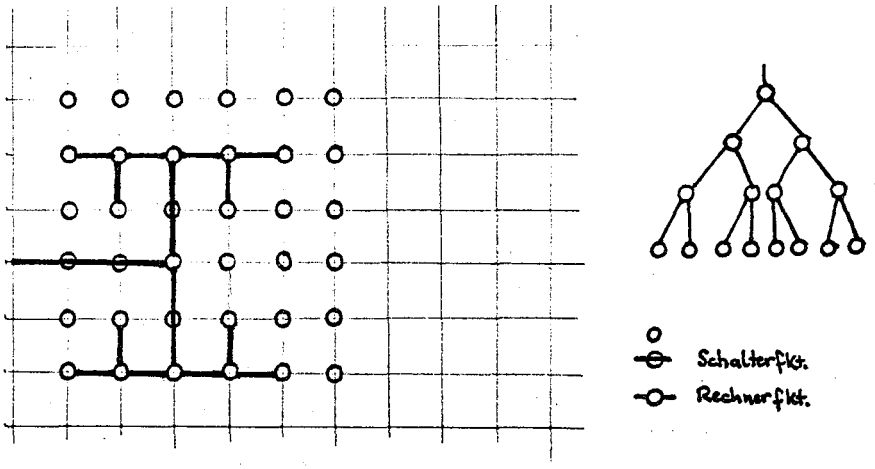


Abb 7.2f Rekonfigurierbares Rechnernetz mit Baumkonfiguration

Baumstrukturen eignen sich besonders, um auf Datensätzen schnelle Datenmanipulationen (Suchen, Vergleichen, Ordnen..) durchzuführen /SON/. Der Aufbau der Baumstruktur kann gleichzeitig auf allen Rechnern geschehen /BEN1/. Jeder Rechner braucht dabei nur einen begrenzten Befehlsvorrat /BEN2/. Weitere Einzelheiten der Rechnerkopplung (Verifikation verteilter Algorithmen, Test und Diagnose der Verbindungsleitungen, etc.) sind in /SEG/ zu finden.

7.3 Schließung der regulären Flächennetze

Die im vorigen Abschnitt eingeführten Flächennetze enthalten bei der Realisierug in VLSI am Rande der behandelten Siliziumfläche sogenannte 'Randeinheiten', also Einheiten mit einer geringeren Zahl von Nachbarn als die Einheiten in der Mitte der Siliziumfläche. Die Zahl der Nachbarn einer Einheit ist hier identisch mit der Kantenzahl pro Einheit im Verbindungsgraphen. Da nach Satz 5.1 von /HAR/ mit der minimalen Zahl d_{min} von Kanten eines Knotens im Graphen G für den Knotenzusammenhang k(G) gilt

$$k(G) \leq d_{min(G)} \qquad (*)$$

ist durch die Randeinheiten der Knotenzusammenhang und damit die t-Diagnostizierbarkeit der Flächennetze (s.7.4) niedriger, als es mit der Netzstruktur möglich wäre. Es liegt deshalb nahe, die Uniformität der Verbindungsstruktur unter den Einheiten 'über den Rand hinaus' beizubehalten,

indem die Einheiten der Ränder geeignet mieinander verbunden werden. Dazu kann man zusätzliche Leiterbahnen (z.B. in einer weiteren 'Sandwich'-Schicht) auf dem Chip aufbringen. Die Existenz dieser Leiterbahnen sei als 'Schließung' bezeichnet.
Sei der Abstand zwischen zwei Einheiten die kleinsten Zahl von Kanten, die eine zusammenhängende Kantenfolge zwischen den Einheiten haben kann.
Für die Schließung der Ränder soll bei den Netzen aus Abb.7.2a,b folgendes gelten:

1) Jeder Weg, der aus einer ununterbrochenen Kantenfolge in der selben Richtung besteht ('Gitterlinie' im Netz), führt schließlich in sich selbst zurück (Endliche Zahl von Einheiten).
2) Der maximale Abstand von zwei Randeinheiten auf einer nichtgeschlossenen Gitterlinie, erhöht um eins, ist die Kantenzahl einer geschlossenen Gitterlinie.

Eine Einheit hat damit einen bestimmten Abstand (Zahl der Kanten) von sich selbst, den 'Umfang' des Netzes in dieser Richtung. Als Einschränkung wollen wir im Folgenden nur Netze betrachten, die bei jeder Einheit in jeder Richtung gleichen Umfang haben. Damit scheiden Strukturen wie in Abb.7.3a aus; Beispiele für die Flächennetze sind in Abb.7.3b,c,d zu sehen.

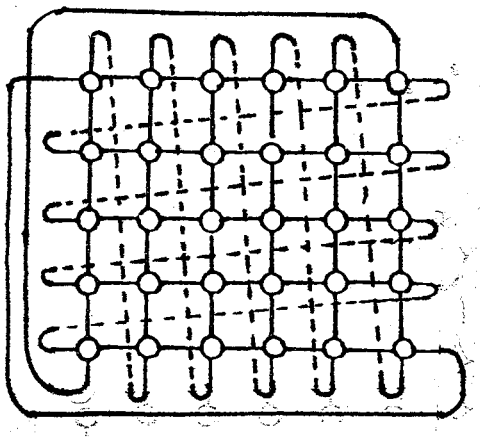


Abb.7.3a Flächennetz mit Torus-ähnlicher Schließung

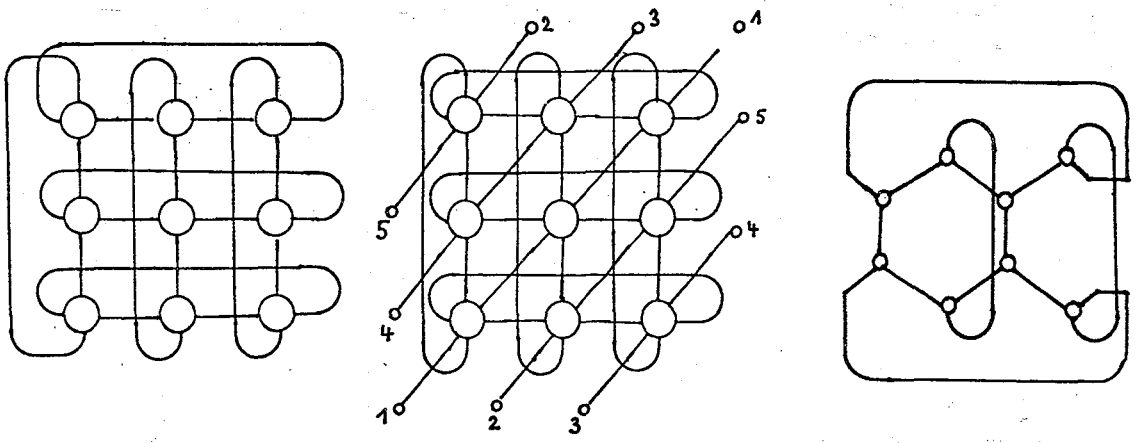


Abb.7.3b,c,d Schließung der regulären Flächennetze

Wie viele Einheiten hat ein solches Netz in Abhängigkeit vom Umfang?
Seien die Netze aus Abb.7.3b,c an zwei verschiedenen Gitterlinien einer Einheit u_0 aufgetrennt (Abb.7.3e,f). Mit gleichem Umfang ist auf allen Gitterlinien auch die gleiche Zahl von Einheiten enthalten.Wenn u_0 eine Randeinheit ist, so sind es u_1 und u_2 ebenfalls. Durch diese Einheiten führen ebenfalls Gitterlinien, so daß auch u_3 als Randeinheit ausgewiesen ist.

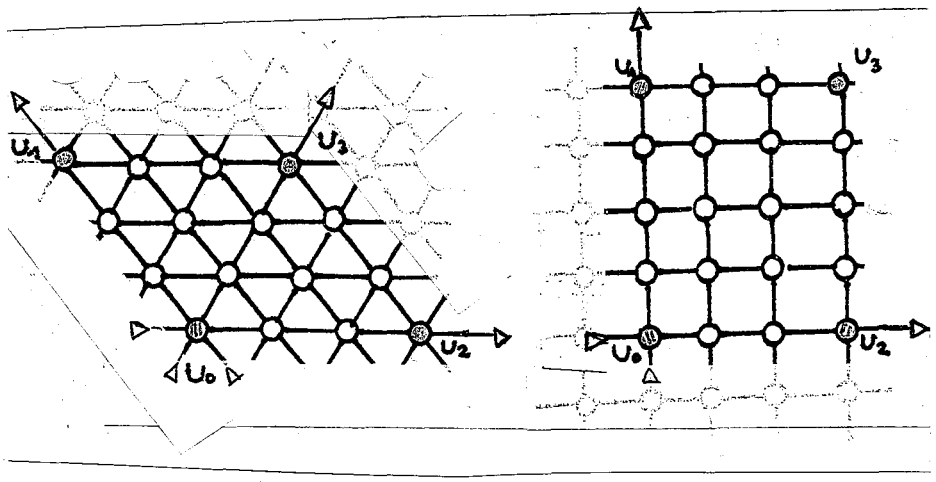


Abb.7.3e,f Flächennetze mit gleichem Umfang

Für den Netztyp mit vier Nachbarn ergibt sich somit die Gesamtzahl der Einheiten als Quadrat der Einheiten auf einer Gitterlinie. Da in Abb.7.3f die

Einheit u_3 auch bezüglich der dritten Gitterlinie durch u_0 eine Randeinheit ist, ist auch für den Netztyp mit sechs Nachbarn die Gesamtzahl der Einheiten im geschlossenen Flächennetz das Quadrat der Zahl der Einheiten auf einer Gitterlinie.

Welchen Zusammenhangsgrad k(G) besitzen die so geschlossenen Flächennetze?
In den Flächennetzen in Abb.7.3b,c,d haben alle Einheiten die gleiche Zahl d(G) Nachbarn. Um einen Knoten abzutrennen, müssen mindestens die d(G) Nachbarn aus G entfernt werden. Da mit dem Entfernen einer kleineren Zahl als d(G) Einheiten weder eine Einheit noch eine größere Menge von Einheiten abgetrennt werden kann, ist bei den regulären, geschlossenen Flächennetzen k(G)=d(G).
Da $d_{min}(G)$ genau dann maximal ist, wenn jeder Knoten die gleiche Kantenzahl $d(G)=|E|/2N$ hat und mit (*) die Relation $k(G) \leq d(G)$ gilt , ist k(G)=d(G) auch der maximale Knotenzusammenhang, der bei einem Graphen mit N Einheiten und $|E|$ Kanten erreicht werden kann.

7.4 t-Diagnostizierbarkeit von regulären, geschlossenen Flächennetzen bei lokalem Testen

Ein wichtiges Problem der oben beschriebenen Rechnernetze ist das Testen und die Diagnose derartig komplexer VLSI-Realisierungen.
Sei die Kommunikationsstruktur des Rechnernetzes durch einen Kommunikationsgraphen G_c dargestellt.
Betrachten wir zunächst Fehlermodelle ohne Verbindungsfehler. Jede Einheit teste nur lokal seine unmittelbaren Nachbarn. Dann wird der Test von einer Einheit an ihren Nachbarn mittels der direkten Kommunikationsverbindung durchgeführt. Die dazu gehörenden Testgraphen G_T sind somit Untergraphen des Kommunikationsgraphen G_c. Je nach Diagnosemodell sind statt der ungerichteten Kommunikationskanten gerichtete Kanten (PMC-Modell und BGM- Modell, Kapitel 2) oder ungerichtete Kanten (Vegleichstestmodell, Kapitel 2.3) im Testgraphen zu finden und es gilt $k(G_T) \leq k(G_c)$.
Was ist die größte Zahl t, so daß diese Testgraphen t-diagnostizierbar ohne Reparatur sind?

Für die zentrale Diagnose (Kapitel 2) wissen wir mit dem Satz (2.1f) von Hakimi und Amin, daß bei dem Knotenzusammenhang $k(G_T)$ des streng zusammenhängenden Testgraphen G_T und den Modellannahmen von Preparata et alii t Fehler mit $t=k(G_T)$ bei $N \geq 2t+1$ Einheiten diagnostiziert werden

können. Werden alle Verbindungen in G_C auch zum Testen in G_T benutzt, so ist der Zusammenhang des Kommunikationsgraphen identisch mit der des Testgraphen: $t=k(G_T)=k(G_C)$.
Damit sind die regulären, geschlossenen Flächennetze mindestens t-diagnostizierbar ohne Reparatur mit $t=k(G_C)$ im PMC-Modell und damit auch im BGM- und Vergleichstest-Modell.
Andererseits gibt es bei $r>k(G_C)$ Defekten Syndrome, die eine eindeutige Diagnose in den Modellen nicht zulassen. Untersuchen wir dazu die Situation, wenn alle Nachbarn einer Einheit u_i bekanntermaßen defekt sind (Abb.7.4a). Das Problem besteht darin, den Zustand von u_i zu diagnostizieren.

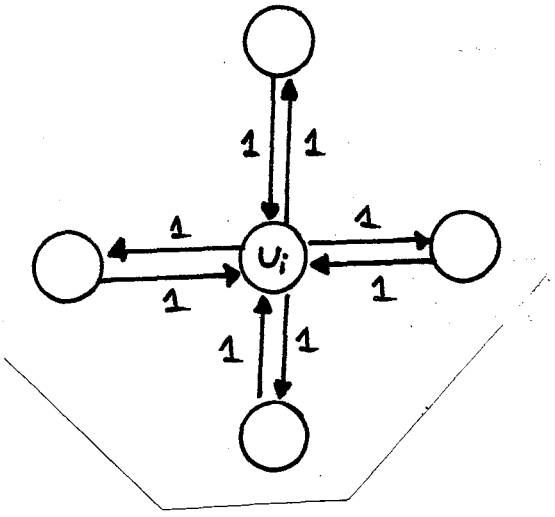


Abb.7.4a Ein nichtdiagnostizierbares Syndrom

Im PMC-Modell läßt sich aus den Testaussagen der defekten Einheiten nichts über u_i erschließen. Auch die Testaussage von u_i über seine Nachbarn ermöglicht keine Aussage über u_i selbst. Das gleiche gilt für das BGM-Modell und das Vergleichstestmodell.
Also sind die regulären Flächennetze nicht nur mindestens, sondern auch höchstens t-Diagnostizierbar mit $t=k(G_C)$.

Die Abhängigkeit dieses Ergebnisses vom Diagnosemodell zeigt das folgende Beispiel. Nehmen wir einmal an, daß alle defekten Einheiten nur einen Fehler haben können (z.B. durch fehlerhafte Produktion gleichartiger Einheiten). Dann finden zwei defekte Einheiten unter dem modifizierten Vergleichstestmodell beim Vergleichstest 'intakt' und die Situation in Abb.7.4a ist eindeutig diagnostizierbar.
Auch bei der dezentralen Diagnose bildet nach Satz 6.1 (Kuhl und Reddy), Satz 6.2a und 6.2b der Knotenzusammenhang des Testgraphen und damit des Kommunikationsgraphen eine obere Schranke für die Diagnostizierbarkeit. Eine Möglichkeit, bei mehr als $k(G_C)$ Fehlern ein System zu diagnostizieren und zu reparieren, ist in Kapitel 9 beschrieben.

8. Lokales Testen und Diagnostizieren im Modell

Wie in Kapitel 7 gezeigt wurde, ist es sinnvoll, bei gekoppelten Rechnern zur Einschränkung der Fehlerfortpflanzung nur beschränkte Kommunikation zuzulassen. Im folgenden Abschnitt wird näher betrachtet, wie parallel zur Bearbeitung einer Aufgabe durch das Rechnernetz eine Fehlererkennung und -lokalisierung stattfinden kann. Je nach Systemvorraussetzungen kann dies durch einen zentral ausgeführten Algorithmus einer zentralen Service-Einheit oder mit einem dezentralen Algorithmus von jeder Einheit geschehen.
Ein wichtiger Vorteil des nachfolgend beschriebenen Diagnoseverfahrens ist seine konzeptionelle Unabhängigkeit vom Testgraphen und damit von der Struktur des verwendeten Rechnernetzes, so daß bei einer Änderung der Netzstruktur (Ausfall, Rekonfiguration, Erweiterung um zusätzliche Einheiten) keine Änderung des Algorithmus nötig ist.

8.1 Beschreibung des Modells

Sei ein Multi-Microcomputersystem MC aus N Einheiten $u_1,\ldots,u_N$ gegeben, das einen Kundentask KT abarbeiten soll.

Dabei sollen folgende Tests zur Fehlererkennung führen:

a) begleitende Tests,
die während der normalen Programmausführung ablaufen. Dies sind alle aus der Code- bzw Datenredundanz resultierenden Überprüfungen wie

- Range-check der Variablen und der Indices
- Kontrolle der Speicherzugriffsrechte /JON/
- Kontrolle der Datentypen und Objektformate /JON/,/RAT/.
- Laufzeitkontrolle der Prozessorfunktion durch redundante Hardware (master-checker /GEY/).
- Paralle, multiple Ausführung des KT durch redundante Hardware (andere Einheiten)/FAR/,/WEN/,/HOP/,/ATT/.

- Kryptografische Codierung und Decodierung der Ein/Ausgabe /SCH/, z.B. Parity, CRC,... /KAF/.

- Benutzerabhängige Plausibilitätskontrolle der benutzten Daten /BON/

- Kontrolle der maximalen Ausführungszeit aller Routinen mittels Zeituhr (watch-dog timer). Dies ist sinnvoll zur Vermeidung von Synchronisations-deadlocks und zur Messung der Benutzerzeit pro Job bei multi-user Betrieb./DAV/

Um die Fehleranhäufung in wenig benutzten Schaltkreisen bzw von wenig benutzten Prozessorinstruktionen zu verhindern /DE/, sollen außerdem periodisch ablaufende, expliziete, überdeckende Tests durchgeführt werden:

b) periodische Hardwaretests

1. Vor Beginn eines neuen KT werden alle dafür benutzten Einheiten getestet.
2. In der Zeit, in der eine Einheit nicht benutzt wird, soll ein Selbsttest ablaufen. Dabei wird nach dem Ende des Selbsttests ein watch-dog timer zurückgestellt.
3. Falls eine Einheit längere Zeit für den KT benutzt wird, soll beim Auslaufen des Timers der Job an das Betriebssystem zurückgegeben (und evtl einer anderen Einheit übertragen) werden. Dann läuft wieder ein Selbsttest ab. Dies garantiert ein maximales Testintervall TI_1 für Selbsttests.
4. Nach einem Intervall TI_2 soll zusätzlich ein Nachbar getestet werden.

Die durch die begleitenden und periodischen Tests entdeckten Fehler sollen folgendermaßen behandelt werden:

Sei die wie folgt beschriebene, initiale Testphase die Teststufe Null. Grundsätzlich wird der fragliche Test erneut ausgeführt, um sicherzustellen, daß es sich um einen dauerhaften Fehler handelt. Bei Datenfehlern (Übertragungsfehler) wird erst ein neuer Datensatz angefordert und der Test erneut ausgeführt, ehe auf Dauerfehler erkannt wird.
Liegt nur ein transienter Fehler vor, so wird ein entsprechender Eintrag in

eine Fehlerfrequenzliste gemacht.
Ist dagegen ein Dauerfehler (z.B. stuck_at_one/zero) aufgetreten, so können beide, Tester bzw. Getesteter, defekt sein. Um dies zu überprüfen, leitet diejenige Einheit u_i, die einen Fehler bei einer anderen findet, zuerst einen Selbsttest ein (s./JON/). Tritt hierbei ein Fehler auf, so geht der Prozessor u_i sofort (oder nach Ausgabe einer Fehlermeldung an alle Nachbarn) in einen HALT-Zustand und löst damit beim ersten Nachbarn u_k, der ihn im Intervall TI_2 testet, eine Fehlermeldung und anschließenden Selbsttest von u_k aus.
Verläuft dagegen der Selbsttest von u_i ohne eine Fehlererkennung, so beginnt die erste Teststufe. Da nach dem bisher erörterten Modell nicht feststeht, ob die Einheit defekt ist, bei der ein Nachbar einen Fehler findet oder der Nachbar selbst, ist es bis zu einem gewissen Grad (vgl.8.2.2) sinnvoll, zur Diagnose nun weitere Einheiten der Nachbarschaft hinzuzuziehen. Dies geschieht dadurch, daß die fehlerentdeckende Einheit diese Nachbarn testet. Dann wird geprüft, ob eine weitere Testausweitung sinnvoll ist, indem der Zustand des Testverfahrens mit einer vorgegebene Abbruchbedingung verglichen wird. Ist die Bedingung nicht erfüllt, so erfolgt ein weiterer Testschritt.
In der zweiten Teststufe testen alle getesteten Nachbarn ebenfalls alle Ihre Nachbarn, und so fort, wie beschrieben.
Dabei wird das System, das den KT bearbeitet, um die Zahl der für den Test benötigten Einheiten verringert, indem die betroffenen Rechner ihre augenblickliche Teilaufgabe unerledigt an ihren Auftraggeber zurückgeben. Dieser ist durch den verwendeten Schedulingalgorithmus festgelegt (s./TIL/).
Ist nun die Abbruchbedingung der Testausweitung erfüllt, so wird nach der gefundenen Diagnose repariert (Computernetzwerke) oder rekonfiguriert (VLSI-Netze), vgl./HAY/.
Nach der Diagnose und Reparatur durch den Testalgorithmus werden die intakten Einheiten wieder für den KT frei.
In den nächsten Abschnitten wird der Diagnosealgorithmus in zentraler und dezentraler Version vorgestellt.

8.2 Lokales Testen mit zentraler, iterativer Bayesdiagnose

Die in den Abschnitten 2 - 5 vorgestellten Diagnoseverfahren beruhen ausschließlich auf dem Modell der zentralen Diagnose, wie sie durch Service-Personal oder spezielle Service-Prozessoren (s.6.0) ausgeführt wird. Im Folgenden soll ein Test-und Diagnoseverfahren mit dieser Voraussetzung erörtert werden.

Dazu wird die iterative, zentrale Bayesdiagnose aus Abschnitt 5.3 mit dem lokalen Testen des Abschnitts 7.1 kombiniert.
Seien in einem System N Einheiten gegeben, die bezüglich ihrer Kommunikationsverbindungen durch den ungerichteten Graph $G_c=(V_c,E_c)$ charakterisiert werden. Angenommen, ein Fehler ist erkannt worden, wie in Abschnitt 8.1 beschrieben. T_0 enthalte nur den Test, der zur Fehlererkennung führte. Dann läßt sich folgender Algorithmus formulieren:

```
LBAY0:
    Der Testgraph G=(V,E) enthalte initial nur die eine Einheit,
    die den Fehler erkannte, und keine Kante.
    n:=0; F:={};
      REPEAT
          n:=n+1;
          Erweitere Testgraph G=(V,E) auf n-te Stufe:
            Teste bisher nicht getestete Nachbarn in V_c der Einheiten aus V
            und erweitere den Testgraphen G um die Nachbarn und die Tests.
            Die Menge aller Tests ist das Syndrom S.
          Bilde F:=F U {alle Einheiten der n-ten Stufe};
          Errechne die Risiken r_i(S) aller Fehlerklassen F_i cF;
          erweitern:=false;
          REPEAT
            Führe eine Bayes-Diagnose aus:
            Suche F_k so, daß r_k(S)= min r_i(S)
                                      F_i cF
            F:=F-F_k;
            IF ( Abbruchbedingung(n) erfüllt ) OR (|V|=N)
               THEN  Repariere nach F_k; Bilde T_0 nach Kapitel 5.3;
               ELSE  erweitern:=true;
          UNTIL (T_0=Ø) OR (erweitern=true) OR (F=Ø)
      UNTIL (T_0=Ø)
```

Der Algorithmus besteht aus 2 REPEAT-Schleifen.
Die äußere Schleife beschreibt das Erweitern des Testgraphen, falls das Abbruchkriterium von dem gerade betrachteten Testgraphen nicht erfüllt ist.
Die innere REPEAT-Schleife führt die iterative Diagnose nach 5.4 durch. Sie wird ebenfalls zugunsten einer Ausweitung des Testgraphen abgebrochen, wenn die erwarteten Testkosten die Reparaturkosten überschreiten.
Dazu gilt:

BEHAUPTUNG 8.1:

1) Der Algorithmus terminiert nach endlicher Schrittzahl.

2) Wenn der Algorithmus terminiert, ist der ursächliche Defekt behoben.

3) Wenn die angewendeten Tests vollständig sind (Bed.2.1a), so ist das System nach Ausführung des Algorithmus fehlerfrei.

BEWEIS:

1) Die iterative Diagnose, und damit auch die innere REPEAT-Schleife, terminiert spätestens dann, wenn alle Einheiten ersetzt sind. Die Ausweitung des Testgraphen, also die äußere REPEAT-Schleife, terminiert spätestens dann, wenn alle Einheiten des Netzes in die Diagnose einbezogen sind.

2) Nach Voraussetzung ist der Test T_0 fehlererkennend für den ursächlichen Defekt. Die Ausweitung des Testgraphen wird abgebrochen, wenn in T_0 kein Fehler mehr vorhanden ist, also auch nicht der ursprüngliche Fehler. Q.E.D.

3) Bei der Vollständigkeit der Tests ist nach Feststellung 6.1 T_0 nur leer, wenn kein Fehler mehr vorhanden ist.

Die in LBAYO benutzte Bayesdiagnose hat eine hohe (Laufzeit)komplexität von maximal $O(2^{2N}(2N+M))$ bei N Einheiten und M Tests (s.Anhang A). Bei der lokalen Diagnose müssen ursprünglich aber nicht alle N Einheiten, sondern nur wenige Nachbarn diagnostiziert werden. Dies läßt sich ausnutzen, um die Diagnosedauer zu erniedrigen, wenn die folgende Frage geklärt ist:
Wann ist die Bayesdiagnose anwendbar?
Die Bayesdiagnose sucht bei einer festen Anzahl N' von Einheiten diejenige Fehlerklasse, die das kleinste Risiko verursacht (s.Abschnitt 5.2). Das Risiko wird dabei unter Benutzung der Wahrscheinlichkeit $P(S/F_i)P(F_i)$ berechnet und ist nur dann definiert, wenn auch ein Syndrom S definiert ist. Definitionsvoraussetzung der Bayesdiagnose ist mithin die Existenz eines Tests zwischen zwei Einheiten und damit impliziet mindestens zwei Einheiten.

Die Bayesdiagnose ist also bei $N' \geq 2$ Einheiten und $|S| \geq 1$ Tests definiert. Der Algorithmus LBAYO wird eingeleitet, wenn das Testergebnis einer Einheit bei einer anderen das Resultat 'defekt' ergab, so daß bei Beginn

von LBAY0 bereits die Definitionsvoraussetzungen für die Bayesdiagnose vorliegen.

Definiert man nun N'(n):=N'(n-1) + (Zahl der Einheiten der n-ten Erweiterung des Testgraphen), so ist mit N'(0):=1, N'(n)$\leq$N für jede feste Anzahl N'(n) eine Bayesdiagnose definiert, deren Diagnosedauer bei kleinem n gering ist trotz der Komplexität

$$O\left(\sum_{i=1}^{n} 2^{2N'(i)}(2N'(i)+M(i))\right)$$

die alle Diagnosezeiten der vorher durchlaufenen n-1 Teststufen enthält und damit größer ist als die der einmaligen Bayesdiagnose mit $O(2^{2N'(n)}(2N'(n)+M(n)))$.
Der Algorithmus LBAY0 führt für die Folge der Testgraphen G(1), G(2),..., G(n) mit der Eigenschaft G(n-1)cG(n) jeweils im n-ten Schritt, also für den Testgraphen G(n):=(V(n),E(n)) mit |V(n)|=N'(n) Einheiten, eine Bayesdiagnose aus und prüft, ob die Abbruchbedingung erfüllt ist. Die Diagnose selbst ist zentral; sie kann von einem extra vorhandenen Service-Prozessor ausgeführt werden und benutzt die lokal ermittelten Testergebnisse, die die einzelnen Einheiten unabhängig voeinander bei ihren Nachbarn erzielt haben.

8.2.1 Die Teststufen in speziellen Testgraphen

Die im Algorithmus LBAY0 beschriebene Ausweitung der Testgraphen soll an einigen Graphen genauer beschrieben werden.

a) Lineare Netze

Betrachten wir eine einfache Kettenstruktur, wie sie Teil eines Ringes sein kann.

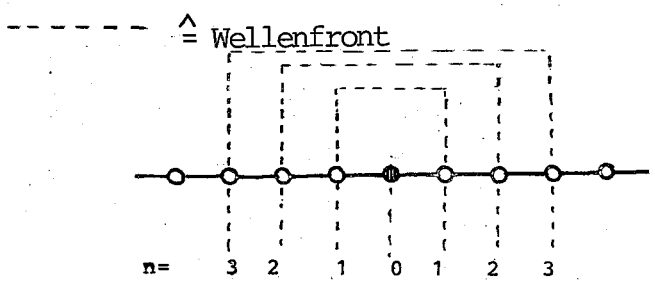


Abb.8.2a Teststufen in einer Kettenstruktur

Die gestrichelten Linien verbinden die Einheiten im gleichen Testzustand ('Wellenfront'). Bei jeder Teststufe kommen zwei Einheiten mit 4 Tests zum betrachteten Testgraphen hinzu; die Zahl der Einheiten ist in der n-ten Stufe somit $N'(n)=1+2n$ bei $n\geq 0$ und die Zahl der Tests $M(0)=0$ und $M(n)=4n-2$ bei $n>0$. Damit hat LBAY0 die Komplexität $O(2^{4n}(15n-1))$.

b) D_{1t}-Testgraphen

Betrachten wir nun die Folge von Teilgraphen $G'(1),G'(2),\ldots,G'(n)$ des in Abschnitt 2.1 eingeführten, gerichteten optimalen Testgraphen. Die D_{1t} Graphen enthalten jeweils 2t Testverbindungen pro Einheit: t Kanten gehen von jeder Einheit aus und t Kanten erreichen jede Einheit. Außerdem sind die Einheiten so angeordnet, daß Einheit u_i die Einheit $u_{i+1},\ldots,u_{i+t}(\text{mod } N)$ testet.

Sei u_1 die Einheit, die den Testalgorithmus auslöst. In der ersten Stufe testet die Einheit u_1 t weitere Einheiten mit t Tests. Also ist $N'(1)=t+1$, $M(1)=t$. In der zweiten Stufe testet jede dieser Einheiten t andere. Trotzdem nimmt die Menge der testenden Einheiten nur um t zu, da alle bisher nicht getesteten Einheiten, die in der zweiten Stufe getestet werden, auch von u_{i+t} getestet werden. Somit ist $N'(2)=2t+1$, $M(2)=t^2+t$. Da bei $n=2$ bereits $2t+1=N$ Einheiten im Testgraphen $G'(2)$ enthalten sind, nimmt bei $n=3$ die Zahl der Einheiten nicht weiter zu, sondern nur die Zahl der Tests. Also ergibt sich $N'(n)=1+nt$ für $0\leq n<3$ und $M(n)=(n-1)t^2+t$ für $0<n\leq 3$. Bei dem nichtoptimalen $D_{1A}(N)$-Design gilt dies natürlich für entsprechend erweiterte Bereiche von n.

c) Reguläre Flächennetze

Behauptung:

Bei den regulären, geschlossenen Flächennetzen nach 7.2 mit $k(G)=3,4,6$ ist

$$N'(n) = 1 + (k(G)/2)(n^2+n) \qquad \text{bei } n1\geq n\geq 0$$
$$M(n) = k(G)\, N'(n-1) \qquad \text{bei } n1\geq n>0$$
$$M(0) = 0 \qquad \text{mit } n1:=(N^{1/2}-1)/2$$

Beweis:

Betrachten wir in Abb.8.2.1b,c und d die Teilnetze, die zwischen den Geraden a und b liegen, inklusive der Einheiten auf der Geraden a. In der

0-ten Teststufe ist jeweils eine Einheit und kein Test vorhanden. Bei n>0 werden in der n-ten Teststufe jeweils n Einheiten zusätzlich aktiviert. Da jedes Flächennetz aus k(G) solcher Teilnetze besteht, werden bei jeder Teststufe k(G)n Einheiten in den Testgraphen einbezogen, insgesamt also

$$N'(n) = 1+\sum_{i=1}^{n} k(G)i = 1+\frac{k(G)}{2}(n^2+n)$$

Da die in einer Teststufe aktivierten Einheiten in dieser Stufe noch nicht selbst testen, gibt es in der n-ten Teststufe N'(n-1) testende Einheiten, von denen jede k(G) Nachbarn testet. Damit gibt es bei n>0 M(n)= N'(n-1)k(G) Tests im System. Bei den regulären, geschlossenen Flächennetzen ist der Umfang des Netzes in jeder Richtung gleich groß (s.7.3). Da bei N Einheiten im System $N^{1/2}$ Einheiten auf einer Gitterlinie sich befinden, sind bei 1+2n aktivierten Einheiten pro Gitterlinie im n-ten Schritt (s.8.2.1a) maximal n_1 Testschritte möglich mit $1+2n_1 \leq N^{1/2}$. Also muß $(N^{1/2}-1)/2 \geq n_1$ gelten.

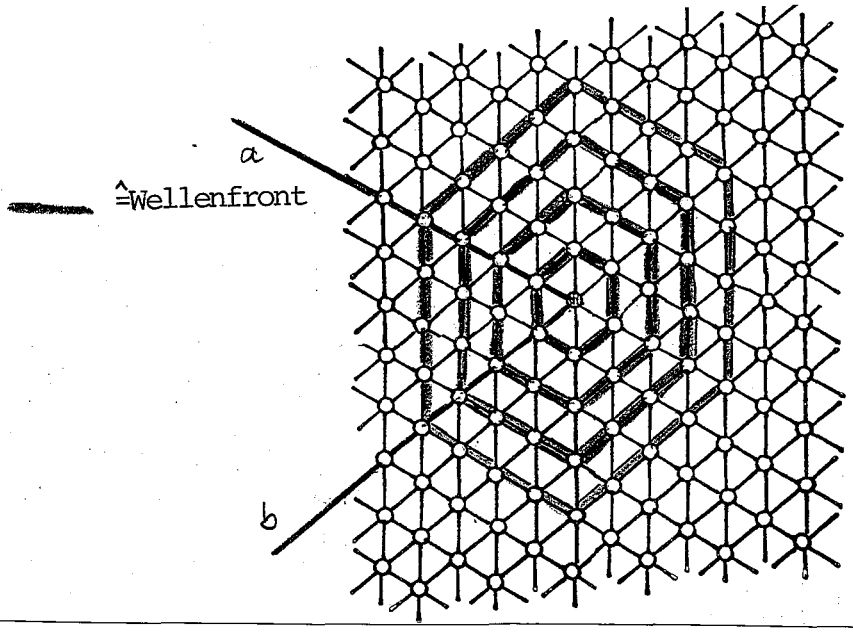


Abb.8.2.1b Teststufen im Flächennetz mit k(G)=6

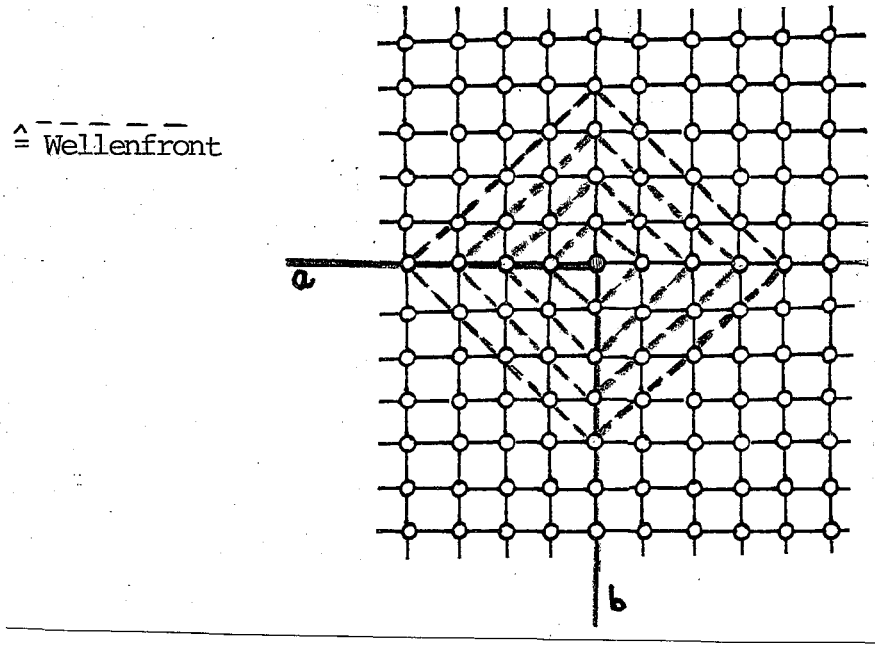


Abb.8.2.1c Teststufen im Flächennetz mit k(G)=4

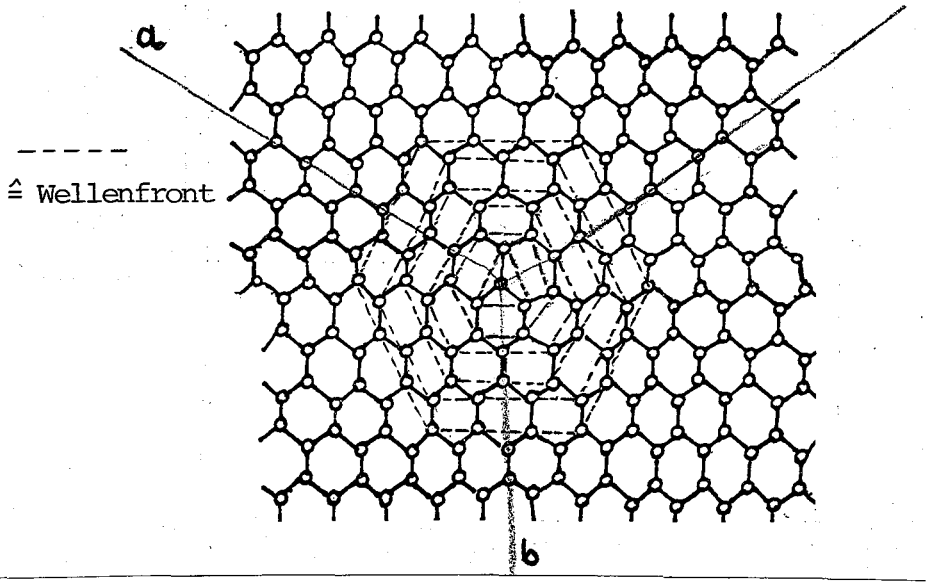


Abb.8.2.1d Teststufen im Flächennetz mit k(G)=3

Damit ist die Komplexität von LBAY0 bei n Schritten von der Ordnung

$$O\Big(\sum_{i=1}^{n} 2^{2(1+k/2(i^2+i))}(2(1+k/2(i^2+i))+k+\frac{k^2}{2}(i^2-i))\Big)$$

mit k:=k(G), $n \leq n_1$.

Eine obere Abschätzung erhalten wir, wenn jeder Summand durch den größten Summanden bei i=n ersetzt wird:

$$O(\ 4n\ (2^k)^{n^2+n}\ (n^2(k+k^2/2) + n(k-k^2/2) +2 +k)\ )$$

und somit

$$O(4(2^k)^{n^2+n}\ (n^3a+n^2b+nc)\ )$$

mit $a:=k+k^2/2$, $b:=k-k^2/2$, $c:=k+2$.

Beispiel:

Betrachten wir das reguläre Flächennetz mit k(G)=4.

Auf einem typischen Mikrocomputer (s.Anhang A) benötigen 2^{10} der betrachteten Operationen 1 sec. Also hat LBAYO bei n=1 mit N'(1)=5 Einheiten und M(1)=4 Tests eine Laufzeit von ungefähr 288000 Operationen oder 288 Sekunden bzw. 5 Minuten. Bei n=2 sind bereits N'(2)=13 Einheiten und M(2)=20 Tests involviert, so zur Diagnose ungefähr 6 Milliarden Operationen oder 70 Tage benötigt werden. Für n=3 mit N'(3)=25 Einheiten und M(3)=52 Tests ist in diesem System eine Diagnose unrealistisch: sie benötigt $34{,}4*10^{16}$ Operationen oder 10 Millionen Jahre.

An diesem Beispiel wird deutlich, daß die Bayesdiagnose und damit auch LBAYO nur bei einer kleinen Anzahl von Einheiten, also bei kleinem n angewendet werden kann. Für große Netze ist damit nur die lokale Diagnose praktikabel.

An diesem Beispiel wird deutlich, daß die Bayesdiagnose und damit auch LBAYO nur bei einer kleinen Anzahl von Einheiten, also bei kleinem n angewendet werden kann. Für große Netze ist damit nur die lokale Diagnose praktikabel.

Betrachten wir die Bayesdiagnose mit 0-1 Loss, also die probabilistische Diagnose, so ist die Komplexität geringer und zwar

$$O(\ (2^{k/2})^{n^2+n}(n^3a\ +n^2b\ +nc)\ )\qquad \text{mit}\quad a:=k+k^2,\ b:=k-k^2,\ c:=2k+2$$

Auch die Größenordung der Ausführungszeiten sind kleiner: Für das Beispiel dauert eine prob.Diagnose bei n=1 0.2 sec, bei n=2 9 Minuten und bei n=3 89 Tage.

Betrachtet man die Systeme nur unter der Annahme, daß nicht mehr als t Einheiten defekt sein können, so läßt sich als Diagnose auch die deterministische Diagnose aus 5.1a verwenden. Legt man außerdem die Annahmen des Vergleichstest-Modells aus 2.3 zu Grunde, so hat der lokale Diagnosealgorithmus bei der Komplexität $O(N^2)$ der systemweiten Vergleichstest-Diagnose die Komplexität

$$O\Big(\sum_{i=1}^{n} N'(i)^2\Big) = O(\ n^5a\ +n^4b\ +n^3c\ +n^2d\ +ne\ ) \quad \text{mit} \quad n\leq n_1, k:=k(G)$$

$e:=1+k(2/3)\ +k^2(1/6\ -1/120)$
$d:=\ k+k^2/8 \qquad b:=k^2/4$
$c:=\ k/6\ +k^25/12 \quad a:=k^2/20$

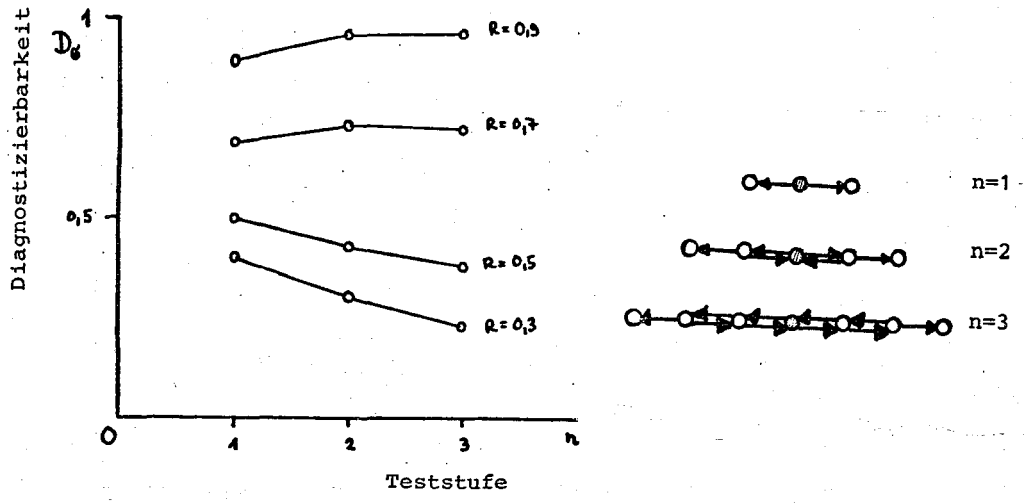
Die Größenordnung der Ausführungszeiten für das Beispiel liegen hier für n=1 bei 26 msec, für n=3 bei 1.24 sec und selbst für n=10 nur 5.1 Minuten.

8.2.2 Beispiele und Ergebnisse

Bei der Einführung von 'Teststufen' stellt sich beim Algorithmus LBAYO die Frage: Was ist ein sinnvolles Abbruchkriterium?
Betrachten wir die verschiedenen Diagnosearten.

a) probabilistische Diagnose

Als Gütemaß für die probabilistische Diagnose sei die Diagnostizierbartkeit des Testgraphen (s.Abschnitt 5.1) betrachtet. Für eine einfache, lineare Kettenstruktur (s.Abb.8.2.2a) ist die Diagnostizierbarkeit der Testgraphen verschiedener Teststufen in Abb.8.2.2a aufgetragen. Zu Grunde lagen vollständige Tests und p=1/2.



Diagnostizierbarkeit Testgraphen

Abb. 8.2.2a Diagnostizierbarkeit der Kettenstruktur

Wie man sieht, ist die Annahme einer besseren Diagnose bei Hinzunahme weiterer Einheiten nicht uneingeschränkt richtig. Bei geringer Zuverlässigkeit R der Einheiten (R=0.3,0.5) nimmt die Wahrscheinlichkeit einer richtigen Diagnose mit der Erweiterung des Testgraphen um unzuverlässige Einheiten ab statt zu, da die fehlende Korrelation der Testergebnisse mit dem Zustand der getesteten Einheit bei testenden, defekten Einheiten die Diagnose erschwert. Aber selbst bei R=0.7 und 0.9 stellt sich bald ein Sättigungseffekt ein; trotz einer Zunahme der Diagnostizierbarkeit von n=1 zu n=2 um rund 5 Prozent nimmt die

Diagnostizierbarkeit des jeweiligen Testgraphen von n=2 zu n=3 bei R=0.9 bereits um 0.02 Prozent ab. Für die einfache Kettenstruktur läßt sich also als gutes Abbruchkriterium unter der prob. Diagnose formulieren, bei n=2 abzubrechen. Eine genauere Analyse der Abhängigkeit der Diagnostizierbarkeit von der Struktur des Testgraphen und der Zuverlässigkeit der Tests und der Einheiten soll Gegenstand späterer Untersuchungen sein.

b) Bayesdiagnose

Bei der probabilistischen Diagnose zeigte sich, daß die Wahrscheinlichkeit der richtigen Diagnose bei Hinzunahme von Einheiten zum aktuellen Testgraphen unter bestimmten Umständen ansteigt.

Gilt dies auch äquivalent für die Kosten?

Betrachten wir wieder die einfache Kettenstruktur. Das Gesamtrisiko R_B der Bayesdiagnose ist durch die über alle Syndrome summierten Einzelrisiken $r_\sigma(S)$ der Bayesdiagnose $\sigma(S)$ gegeben. In Abbildung 8.2.2b ist die Entwicklung des Gesamtrisikos in Abhängigkeit der Teststufen aufgetragen. Parameter sind die Reparaturkosten C_r einer Einheit. Der Kostenkoeffizient L_{ik} ist analog zu 5.2 definiert und es werden vollständige Tests (s. Bedingung 2.1a) vorausgesetzt.

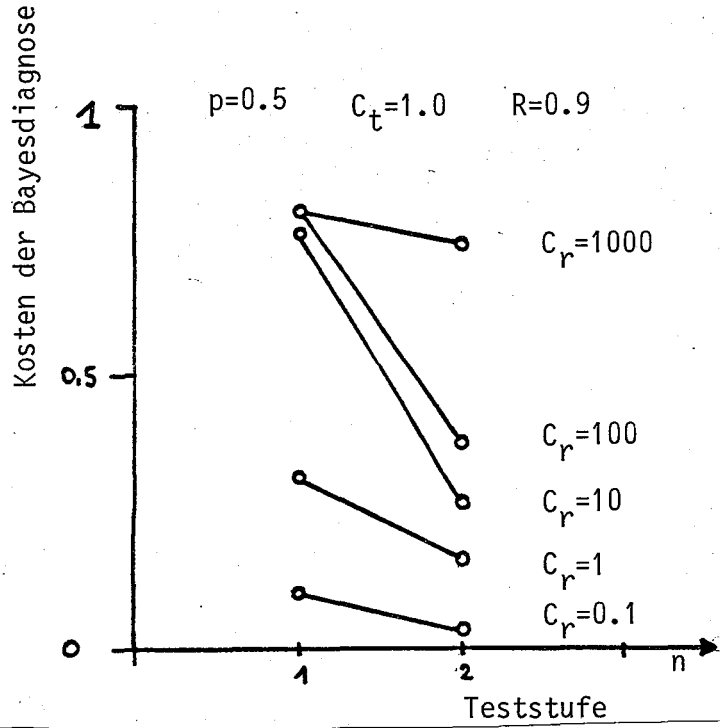


Abb. 8.2.b Risiko der Bayesdiagnose

Auch hier ist im Vergleich mit 8.2.2a erkennbar, daß bei Vergrößerung des Testgraphen durch Einheiten mit großer Zuverlässigkeit (R=0.9) eine Kostenreduzierung unter Umständen möglich ist.
Angenommen, wir wählen als Abbruchbedingung der Testausweitung, daß die erwarteten Reparaturkosten kleiner werden als die veranschlagten Testkosten des erweiterten Testgraphen. Wann wird der Algorithmus abbrechen?
Das Bayes-Risiko einer Diagnose σ bei gegebenen Syndrom S ist nach Abschnitt 4.1

$$r_\sigma(S) = \sum_{i=0}^{2^{N'(n)}-1} L_{i\sigma(S)} P(S,F_i)$$

Versuchen wir nun, eine obere Schranke für das Risiko der Bayesdiagnose zu finden, das bei n=1 entstehen kann. Seien C_t die Kosten, eine Einheit während einer Teststufe zum Testen zu verwenden und C_r die Kosten, eine Einheit zu reparieren. Die Testkosten einer Teststufe (Testgraphen) sind somit $C_T := N'(n)C_t$. Der Kostenkoeffizient L_{ik} sei nach Kapitel 5.2 definiert:

L_{ik} := C_r(Zahl der in F_k, aber nicht in F_i enthaltene Einheiten)
$+ C_T \cdot \Theta_{ik}$

mit

$\Theta_{ik} = \begin{cases} 1 & \text{wenn eine Einheit in } F_i, \text{ aber nicht in } F_k \text{ enthalten ist} \\ 0 & \text{sonst.} \end{cases}$

- Sei $C_r \geq C_t$.
Betrachten wir das Risiko der Fehlerklasse $F_0 = \{\} =$ (keine Einheit defekt). Dann enthält die Fehlerklasse sicher keine Einheit zuviel und es ist

$$r_\sigma(S) \leq r_0(S) = \sum_{i=1}^{2^{N'}-1} N'(1)C_t P(S\ F_i) \leq N'(1)C_t P(S) < N'(1)C_t$$

Wenn C_r wesentlich größer als C_t ist, hat F_0 tatsächlich das kleinste Risiko. Dies ist um so erstaunlicher, als bei vollständigen Tests für fast alle Syndrome $P(S,F_0)=0$ gilt. Bei relativ hohen Reparaturkosten wird bei

der Bayesdiagnose also für eine Fehlerklasse entschieden, die garnicht auftreten kann (!).

- Sei andererseits $C_r < C_t$.
  Betrachten wir das Risiko der Fehlerklasse $F_e := \{u_1, \ldots, u_N\}$ (alle Einheiten defekt). Dann enthält die Fehlerklasse sicher alle defekten Einheiten und es gilt

$$r_\sigma(S) \leq r_e(S) = \sum_{i=0}^{2^{N'}-2} N'(1) C_r P(S\ F_i) \leq N'(1) C_r P(S) < N'(1) C_t$$

In beiden Fällen bricht der Algorithmus in der ersten Teststufe bereits ab. Eine Ausweitung des Testgraphen mit den Testkosten als Abbruchkriterium ist also nicht sinnvoll.

8.3 Lokales Testen und dezentrale Bayesdiagnose

Unter dem in 8.1 vorgestellten Modell soll nun ein dezentraler, lokaler Diagnosealgorithmus formuliert werden.
Jede Einheit führt für sich eine Liste von Testergebnissen ('Syndromtafel') und von Zuständen aller Einheiten, über die Testergebnisse vorliegen ('Systemtafel'). Die Systemtafel einer Einheit enthält mindestens die Zustände der Nachbareinheiten und ist vor der ersten Ausführung des Diagnosealgorithmus mit 'intakt' initialisiert.
Sei ein Fehler entdeckt (s.8.1a,b) und ein initialer Selbsttest durchgeführt worden. Die fehlerentdeckende Einheit testet sodann alle ihre Nachbarn und gibt anschließend einen Testauftrag an diese. Der Testauftrag enthält eine Absendernummer, die Nummer der Testrunde und alle Testergebnisse des Auftraggebers.
Um eine korrekte Absenderidentifizierung zu erreichen, seien die Nachrichten geeignet nach Kapitel 6.2 verschlüsselt.
Der Algorithmus lautet somit für jede Einheit u_i:
LBAY1:

```
receive_message(Nachricht,Sender=Nachbar u_j);
              (* Eine Nachricht vom Nachbar u_j trifft ein*)

CASE Nachricht OF

    Testauftrag:
      IF (das Nachrichtenformat stimmt)
         AND (u_i nicht im Teststatus) AND ( u_j ist 'intakt')
         (* keine wiederholte Aktivierung durch defekte Einheiten *)
         THEN
             Teststatus:=true;
             notiere Testergebnisse des Auftraggebers in der Syndromtafel;
             teste alle Nachbarn einschließlich des Auftraggebers;
             send_message(Testergebnis, Auftraggeber);
             Diagnostiziere (Syndromtafel);
             IF (Abbruchbedingung = erfüllt)
                THEN Terminate          (* Teststatus beenden *)
                ELSE       (* aktiviere alle intakten Nachbarn zum Testen *)
                    send_message(Testauftrag, alle 'intakten' Nachbarn
                                              außer dem Auftraggeber)
```

und setze jeweils einen Timer mit max. Wartezeit.

```
Testergebnis:
  IF     (das Nachrichtenformat stimmt)
         AND (u_i ist Beauftragter oder Auftraggeber)
         AND (die Nachricht ist neu)
              (* falsche oder wiederholte Nachrichten werden ignoriert *)
    THEN
         vervollständige die Syndromtafel;
         send_message(Testergebnis, Auftraggeber und alle Beauftragten
                                                  außer Nachbar u_j)
         setze den Timer von u_j zurück.

Testauftrag beendet:
   IF         (das Nachrichtenformat stimmt)
          AND (dieser Nachbar einen Auftrag erhalten hat)
          AND (u_i im Teststatus ist)
              (* zum Status inkonsistente Nachrichten werden ignoriert *)
     THEN
         Nachbar(u_i):='fertig';
         IF (alle Nachbarn fertig)
            THEN                  (* Teststatus beenden *)
               Diagnostiziere (Syndromtafel);
               Terminate;

Timer ausgelaufen:                (* Beauftragter ist defekt *)
    Testergebnis:=t_ij ='defekt';(*Time-out ist auch ein Testergebnis*)
    vervollständige die Syndromtafel;
    send_message(Testergebnis, Auftraggeber);
    u_j:=fertig;
    IF (alle Nachbarn = fertig)
       THEN       (* Teststatus beenden *)
          Diagnostiziere (Syndromtafel);
          Terminate;
END_CASE;
```

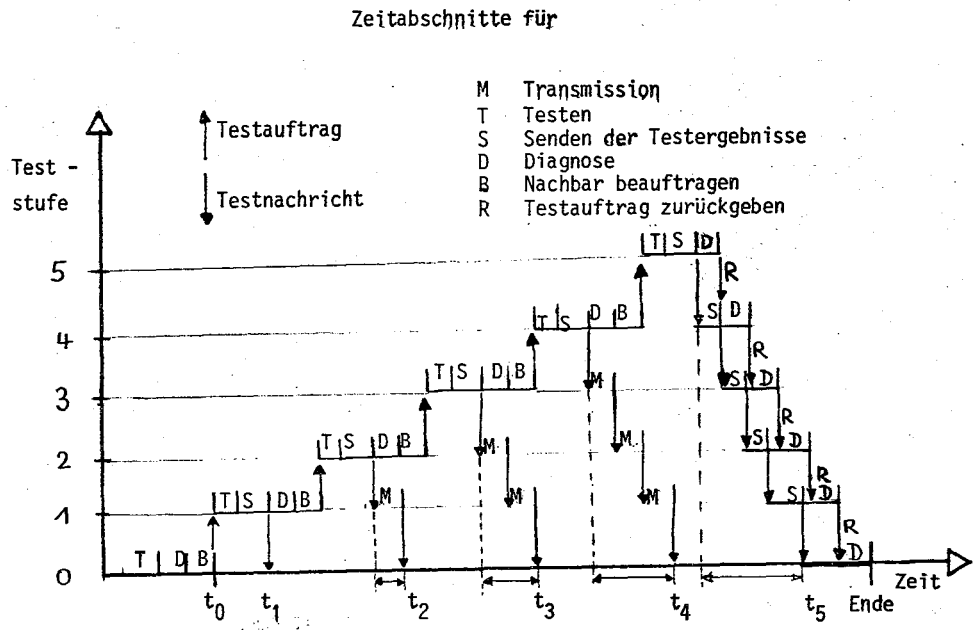
wobei folgende Prozeduren verwendet werden:

Diagnostiziere (Syndromtafel), z.B. Bayesdiagnose:

Bilde F:={ alle Einheiten,
über die Tests in der Syndromtafel vorliegen};
Suche F_k so, daß $r_k(S)=\min_{F_i^i \, cF} r_i(S)$

Terminate:
Ändere die Systemtafel mit F_k;
Gib Testauftrag und Ergebnis an Auftraggeber zurück;
Beende den Teststatus.

Dieser Algorithmus beschreibt die Testaktivierung der jeweiligen Nachbareinheiten. Mit der Systemtafel und einer bekannten Umgebungsstruktur kann jede Einheit eine Diagnose nach Abschnitt 5.1 bzw 5.2 durchführen. Wenn die Wartezeit des Watchdog-Timers lang genug ist, um alle Syndrome an alle involvierten Einheiten weiterzuleiten zu können, werden nach 6.1 in allen intakten Einheiten auch die gleichen Syndrome ausgewertet. Die dezentrale Syndromauswertung bei allen intakten Einheiten führt zur gleichen Systemtafel und, für die lokale Diagnose besonders wichtig, bei gleicher Länge der Testauftragskette auch zur gleichen Entscheidung über Erweiterung des Testgraphen.
Wie lange muß nun eine in der n-ten Teststufe aktivierte Einheit warten, um sicher alle Testergebnisse zu erhalten ?
Jede Einheit verteilt die Testergebnisse nur innerhalb der Auftragskette. Damit kann keine Vermischung der Testergebnisse mit der einer anderen Kette entstehen. Wenn eine Einheit der n-ten Stufe ihre Testergebnisse verschickt, erhält die Einheit der Stufe Null die Nachricht nach n-1 Zeiteinheiten der Transmission und jede andere Einheit der n-ten Stufe die Nachricht nach insgesamt maximal 2n-1 Zeiteinheiten der Transmission. Sind maximal n_{max} Stufen im Testgraphen möglich (z.B. $n_{max}=3$ in D_{1t}-Graphen, s.8.2.1), so gilt diese Zeitobergrenze der Transmission für alle Testteilnehmer. Dazu addiert sich noch die Zeit zur Aktivierung bis zur n-ten Stufe.
Die folgende Skizze soll dies veranschaulichen. Darin sind als Pfeile die Übertragung von Testnachrichten eingetragen. Der zugrunde liegende Testgraph wird nur linear in einer Dimension betrachtet. Für jede Aktivität eines Knotens (Testen, Senden, ...) ist ein Zeitintervall eingetragen, wobei die Intervalle jeweils mit einem Buchstaben bezeichnet sind.



Beispiel eines Aktivierungsablaufs

Wie in der Skizze zu sehen ist, führt die Ausweitung des Testgraphen zu immer größeren Zeitabständen zwischen dem Senden und dem Eintreffen der Testnachrichten beim Auftraggeber. Sei der Diagnosealgorithmus zum Zeitpunkt t_0 gestartet worden. Dann ergibt sich der Zeitpunkt des Eintreffens der Testergebnisse von Stufe n in Stufe 0 zu $t_T=t_0+(n-1)(T+S+D+B+M)+T+S$, also linear in n.

Den Testvorgang kann man sich wie eine Welle vorstellen, die, ausgelöst von einer Einheit, sich nach allen Seiten ausbreitet, um die eigentlich defekte Einheit zu bestimmen, bei Erfüllung des Abbruchkriteriums anhält, wieder zurückflutet und verschwindet. Die Wellenfront wird dabei durch jene Einheiten gebildet, die im gleichen Zeitschritt über eine Ausweitung des Testzustandes entscheiden.

Der Begriff der Wellenfront in der Algorithmik kommt ursprünglich aus dem Gebiet der systolischen Felder (s.7.1) und findet inzwischen auch eine Anwendung bei dezentralem Scheduling /TIL/.

Welche Störungen des Algorithmus können auftreten?

Die Störmöglichkeiten von defekten Einheiten sind

passiv: a) -keine Transmission der Nachrichten

aktiv: b) -Störung der Nachbarn durch dauernde Testaktivierung oder Rückgabe des Testauftrags an falsche Nachbarn;

c) -Fälschung der Ergebnisse;
z.B. Ändern von Testergebnissen bei der Transmission oder Generieren von falschen Testergebnissen.

Dies wird in LBAY1 vermieden durch

a) -Umgehung von 'defekten' Nachbarn im Netz
-Zeitüberwachung der Beauftragten. Bei nichtvollständigen Tests kann ein Beauftragter defekt sein, keine Testergebnisse weiterleiten und der Algorithmus terminiert nicht.

b) -Akzeptierung des Testauftrags durch u_i nur dann, wenn u_i nicht bereits im Teststatus ist.
-Nach Abschluß der Tests ist die defekte Einheit in Systemtafel gekennzeichnet und ihr Testauftrag wird nicht akzeptiert.
-Das Ende des Testauftrags wird nur von den beauftragten Nachbarn angenommen

c) -Codierung der Nachrichten;
Nachrichten von und über Nicht-Nachbarn werden nur akzeptiert, wenn das Format stimmt, also eine unverfälschte Orignalnachricht vorliegt.
-Akzeptierung der Testergebnisse nur von Nachbarn der Testauftragskette

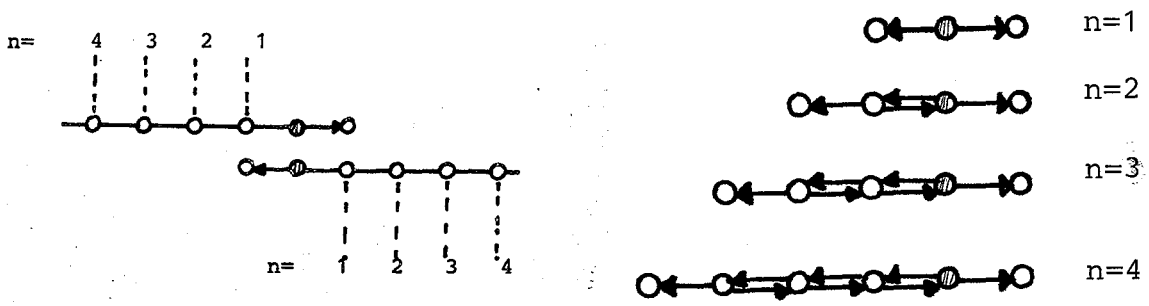
8.3.1 Testausweitung in speziellen Testgraphen

Die Teststufen der dezentralen, lokalen Diagnose unterscheiden sich weitgehend dadurch von denen der zentralen Diagnose von Abschnitt 8.2, daß die Testergebnisse über die Auftragsketten verteilt werden müssen. Nicht jede Einheit erhält somit die Testergebnisse jeder anderen Einheit. Es entstehen

mehrere Testauftragsketten, die bis auf die die Testaktivität auslösende Einheit ('Wurzelknoten') keine Einheiten gemeinsam haben. In symmetrischen Netzen entstehen dabei mehrere, gleichartige Testgraphen.

a) Kettenstruktur

Die in Abb.8.2.1a abgebildete Kettenstruktur wird durch den Testalgorithmus in zwei Aktivitätszweige (s. Abb.8.3.1a) aufgespalten.

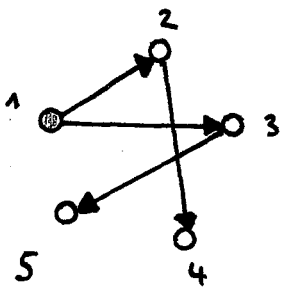


Aktivierung Testgraphen

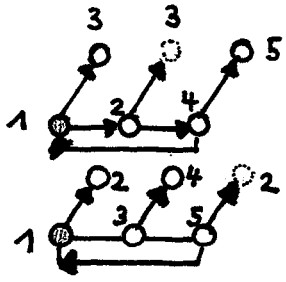
Abb.8.3.1a Aktivierungsgraphen der Kettenstruktur

b) D_{1t}-Graph

Angenommen, ein System gestatte zum Testen einen D_{1t}-Testgraphen nach Abschnitt 2.1. Dann aktiviert LBAY1 wie LBAY0 in 2 Teststufen das gesamte System. Angenommen, jede Einheit u_i aktiviert ihre Nachbarn in der Reihenfolge $u_{i+t}, u_{i+t-1}, \ldots, u_{i+1}$. Dann kann jede aktivierte Einheit nur jeweils ihren Nachbarn u_{i+t} aktivieren, da alle anderen bereits vorher aktiviert sind. Es existieren somit t Auftragsketten, denen t Testgraphen entsprechen. In Abb.8.3.1b und c ist dies am Beispiel des D_{12}-Graphen dargestellt.



Aktivierung des D_{12}-Graphs



Testgraphen des D_{12}-Graphs

b) c)

Abb.8.3.1b,c Aktivierung und Testgraphen des D_{12}-Graphen

Zur Verdeutlichung der Regelmäßigkeit der Testgraphen sind in Abb.8.3.1c für manche Einheiten zwei Knoten eingezeichnet, obwohl die Tests die selbe Einheit testen.

Für die Diagnose muß natürlich auch ein Nachrichtenaustausch in Gegenrichtung der Testrichtung möglich sein.

Der Ablauf des Testalgorithmus LBAY1 im D_{12}-Graphen sei in folgender Tabelle verdeutlicht:

Sei beispielsweise u_2 defekt.

Teststufe/	u_1	u_2	u_3	u_4	u_5
n=0	u_1 testet u_2 und findet 'defekt'. u_1 führt Selbsttest durch; Resultat='intakt'.				
n=1	u_1 testet u_3 und erneut u_2. Beauftragt u_3 und u_2, da das Testergebnis nicht sicher.				

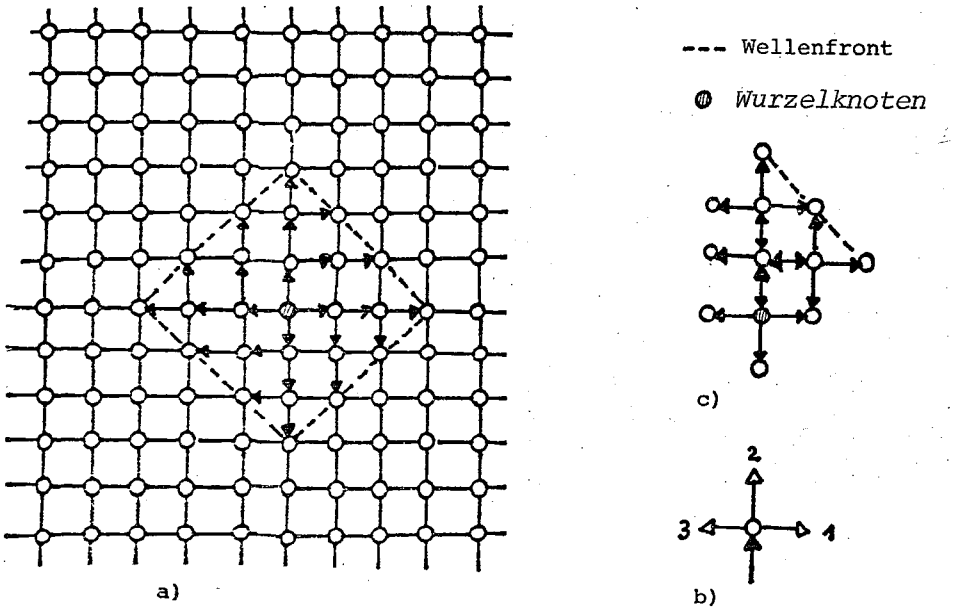
Teststufe/ u_1	u_2	u_3	u_4	u_5
n=2	testet u_4, u_3 sendet t_{24},t_{23} an u_1. Beauftragt u_4,u_3.	testet u_4,u_5 sendet t_{34},t_{35} an u_1. Beauftragt u_4,u_5.		
n=3		akzeptiert keinen Testauftrag von u_2,da schon im Teststatus.	akzeptiert Testauftrag von u_2,aber nicht von u_3, da schon im Teststatus. Testet u_5,u_1 sendet t_{45},t_{41} an u_2	Testet u_1,u_2 sendet t_{51},t_{52} an u_3
	übermittelt t_{41},t_{45} an u_1	übermittelt t_{51},t_{52} an u_1	Abbruchbed. erfüllt. Diagnose. Auftragsende an u_2.	Abbruchbed. erfüllt. Diagnose. Auftragsende an u_3.
n= 2	Akzeptiert 'Auftragsende' von u_4. Diagnose. Sendet 'Auftragsende' an u_1	Akzeptiert 'Auftragsende' von u_5. Diagnose. Sendet 'Auftragsende' an u_1		
n=1	akzeptiert 'Auftragsende' von u_2 und u_3. Diagnose.			

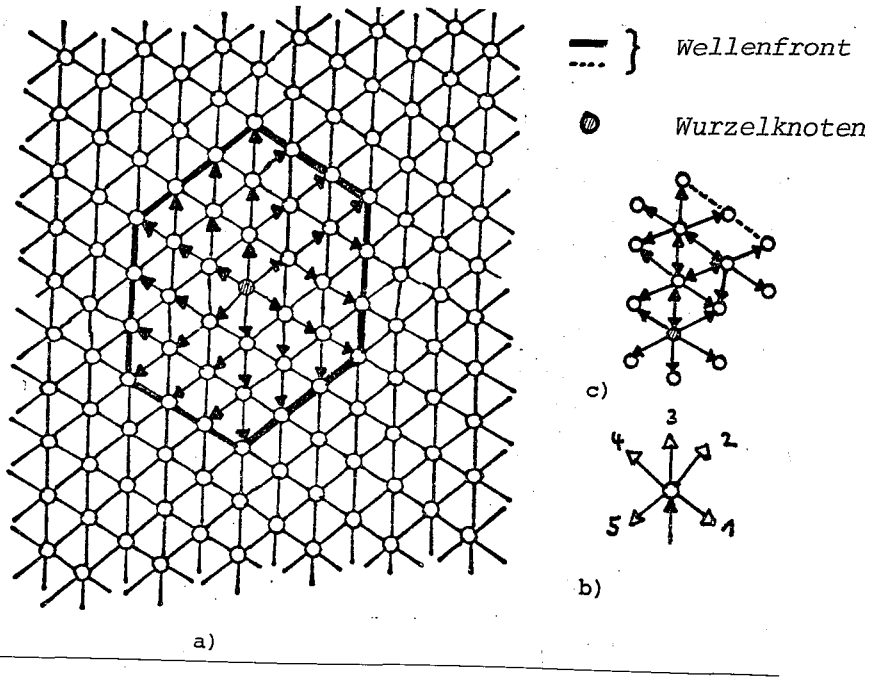
Bei der Diagnose liegen den Einheiten u_3 und u_5 ein Testgraph aus Abb.8.3.1c vor. Da die Testgraphen einen Ring (D_{11}-Testgraphen) als Teilgraph enthalten, sind sie 1-diagnostizierbar.

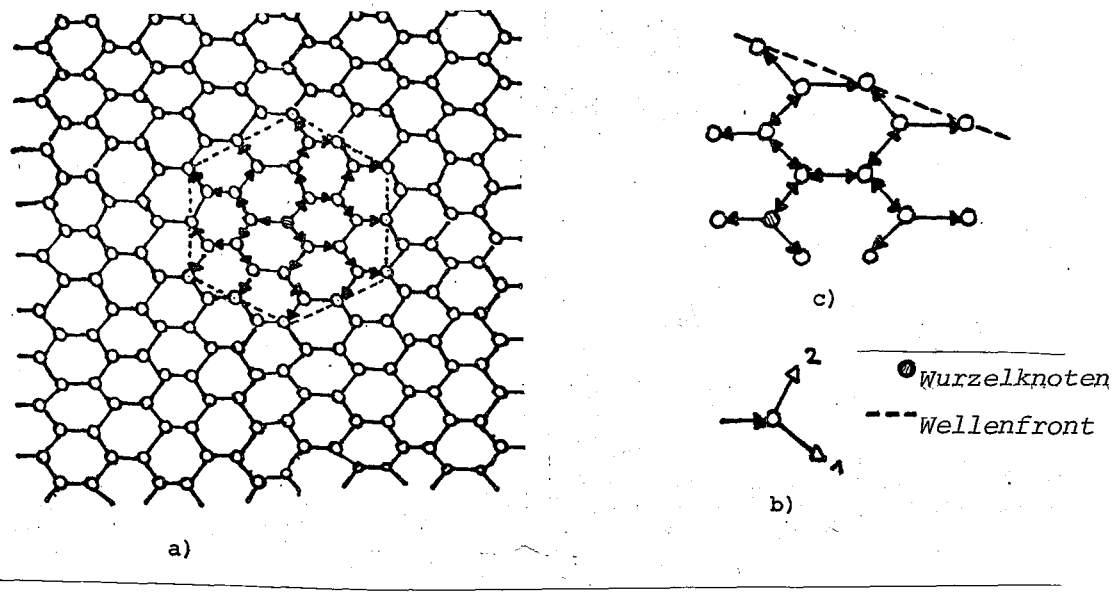
Bei u_1 ist die gesamte Information und damit der komplette Testgraph vorhanden, der 2-diagnostizierbar ist.

c) Flächennetze

Die folgenden Abbildungen sollen einen Eindruck vom Ablauf von LBAY1 vermitteln.





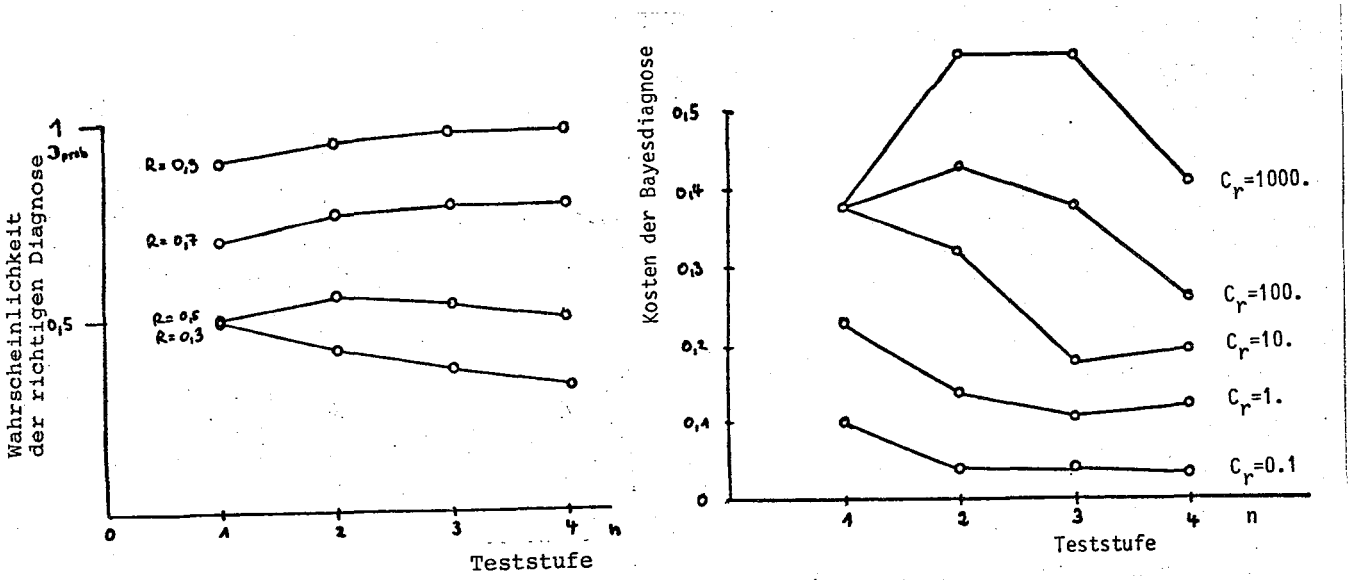


In Abb.a) ist für die verschiedenen Netztypen der Aktivierungszustand für n=3 bzw. n=4 bei k(G)=3 eingezeichnet. Ein Pfeil symbolisiert jeweils eine Aktivierungsrichtung; die Wellenfront wird durch alle in den Testalgorithmus einbezogenen Einheiten gebildet.
Betrachtet man nur die gerichteten Kanten (Pfeile) und die damit verbundenen Einheiten, so erhält man einen gerichteten Graphen mit einer Baumstruktur. Der Wurzelknoten dieser Bäume ist die Einheit, die den Diagnosealgorithmus ausgelöst hat.
In Abb. b) ist das dieser Aktivierung zugrunde liegende, lokale Schema gezeigt: jede aktivierte Einheit aktiviert die Nachbareinheiten im Uhrzeigersinn, also zuerst Nachbar 1, dann Nachbar 2, usw. In Abb.c) ist schließlich der Testgraph gezeigt, der bei n=3 bzw. 4 jedem der k(G) Aktivierungsbäume zugeordnet ist.

8.3.2 Beispiel

Für die probabilistische Diagnose und die Bayesdiagnose gelten qualitativ die in 8.2.2 festgestellten Zusammenhänge.
Betrachten wir das Beispiel der Kettenstruktur aus dem vorigen Abschnitt 8.3.1a. In der folgenden Abbildung ist die Diagnostizierbarkeit und das Risiko der Folge der dezentral durch LBAY1 aktivierten Testgraphen aus Abb.8.3.1a bei probabilistischer und Bayesdiagnose gezeigt, so wie sie sich für alle Einheiten der beiden Auftragsketten darstellt. Für die probabilistische Diagnose wurden vollständige Tests und p=1/2 angenommen sowie außerdem C_t=1.0 und R=0.9 für

die Bayesdiagnose.



a) Diagnostizierbarkeit bei p=1/2

b) Risiko bei C_t=1.0 R=0.9

Abb.8.3.2 Verhalten bei Vergrößerung des Testgraphen

Im Vergleich zu Abschnitt 8.2.2 zeigt sich, daß durch die dezentrale Betrachtung die Menge der Einheiten pro Testgraph langsamer wächst und damit die Diagnostizierbarkeit noch bis n=3 ansteigt. Als Abbruchkriterium bietet sich also hier die 3.Teststufe an.

Das Risiko nimmt bei nicht zu großen Reparaturkosten asymptotisch ab; auch hier bietet sich die 3.Teststufe als Abbruchkriterium an.

9.0 Selbstreparatur von Rechnernetzen

Sei ein System gegeben aus N Einheiten, deren Kommunikation bezüglich des physikalischen Nachrichtenaustauschs durch den ungerichteten Graphen G=(V,E) gegeben ist. Dieser habe den Knotenzusammenhang k(G). Wie im Kapitel 6 und 7 gezeigt wurde, bildet k(G) (in 7.1 die Zahl der Nachbarn) eine obere Schranke t bei einer 1-Schritt Diagnose. In großen Netzen (s.7.1) kann trotz kleiner Zahl der Nachbarn die Gesamtzahl N aller Einheiten des Systems und damit die Zahl der Defekte viel größer als k(G) sein. Deshalb wäre es günstig, ein Verfahren angeben zu können, das eine Diagnose und Reparatur auch bei mehr als k(G) defekten Einheiten gestattet.
In Computernetzen können zusätzliche, redundante Einheiten bei einem Ausfall einer oder mehrerer Einheiten ein Weiterarbeiten des Gesamtsystems ermöglichen. In manchen Anwendungen ,z.B. bei der Matrizenmultiplikation aus Kapitel 7, muß dabei allerdings die Netzstruktur gewahrt bleiben.
Da es in diesen Netzen nicht sinnvoll ist, einen 'pool' von Ersatzeinheiten anzulegen, die an die Stelle von jeder Einheit schaltbar sind (Leitungsprobleme!), empfiehlt es sich, z.B. jede Einheit redundant mit einer Reserveeinheit (kalte Reserve) auszulegen, die im Fehlerfall an die gleiche Stelle treten kann, ohne vorher im Kommunikationsgraphen enthalten zu sein.
Das Problem, wie ein für einen Algorithmus notwendiger Teilgraph auf einem umkonfigurierbaren Netz realisiert werden kann, ist beispielsweise in /HAY/ und in /MAEH2/ behandelt und soll nicht näher betrachtet werden.

9.1 Bedingungen der Selbstreparatur

Sei ein System mit N Einheiten, dem Kommunikationsgraphen G und zusätzlich zuschaltbaren, redundanten Einheiten gegeben.
DEFINITION:

> Ein System mit dem Graphen G heißt 't-Fehler selbstreparierend', wenn nach Ausfall von nicht mehr als t Einheiten
> die Diagnose und Reparatur von Einheiten des Systems selbst durchgeführt wird
> und das System nach endlich vielen Zeitschritten den Kommunikationsgraphen G aufweist mit ausschließlich intakten Einheiten.

Obige Definition impliziert, daß die defekten Einheiten vom System abgeschaltet und die Reserveeinheiten automatisch dazugeschaltet werden. Geschieht dies durch eine zentrale Einheit, so ist das System nur 0-Fehler selbstreparierend,

da bei Ausfall der zentralen Einheit keine Reparatur mehr möglich ist. Eine dezentrale Lösung besteht darin, eine Reparatur einer Einheit (Abschalten und Dazuschalten der Reserveeinheit) durch einen Nachbarn vornehmen zu lassen. Dabei können sich aber Probleme ergeben.

Bei den meisten Systemen nimmt dabei die Diagnose wesentlich mehr Zeit in Anspruch als die eigentliche Reparatur.

Betrachten wir nun folgende Situation.

Angenommen, es ist t defekten Einheiten möglich, eine Reparatur zu veranlassen. Dann können die defekten Einheiten sich gegenseitig beauftragen, die intakten Einheiten ihrer Nachbarschaft abzuschalten.

Betrachten wir eine defekte Einheit u_i,

die nur eine minimale Zahl von Nachbarn hat. Ist die Diagnosedauer länger als die Zeit zum Abschalten aller direkten Nachbarn, so kann die Einheit u_i alle ihre intakten Nachbarn in kürzerer Zeit auswechseln (und damit bezüglich der Diagnose initialisieren), als die intakten Nachbarn die Einheit u_i als defekt diagnostizieren und reparieren können. Damit gibt es eine defekte Einheit, die nie repariert wird und das System bleibt defekt. Abgesehen von den unnötigen Reparaturkosten kann sich somit eine Situation ergeben, in der eine korrekte Selbstreparatur des Netzes nicht mehr möglich ist, d.h. G ist nicht mehr selbstreparierend.

Wenn bei nicht mehr als t Fehlern für eine Reparatur t+1 Einheiten übereinstimmen müssen, muß darunter auch mindestens eine intakte Einheit sein. Damit ist es sinnvoll, folgende Reparaturbedingung einzuführen:

ANNAHME 9.1

Für eine Reparatur (Austausch bzw. Abschalten) einer Einheit müssen mehr als t Einheiten übereinstimmen.

Dies ist beispielsweise mit einem kryptographischen Code möglich, bei dem t+1 Schlüssel notwendig sind, um das richtige Codewort für den Schalter zu errechnen.

Sei wieder die Kardinalität einer Menge V mit v bezeichnet.

SATZ 9.1:

Für ein System mit N>2 Einheiten und dem zusammenhängenden, ungerichteten Kommunikationsgraphen G = (V,E) gelte Annahme 9.1 und es sei t>0. Das System ist t-Fehler selbstreparierend d.u.n.d., wenn

a) es bei beliebiger Anordnung von maximal t Defekten mindestens einen zusammenhängenden Teilgraphen G' von G mit ausschließlich intakten Einheiten und v'>t gibt.

b) jeder auftretende Defekt auch erkannt und eine Diagnose und Reparatur eingeleitet wird

c) ein Diagnosealgorithmus existiert, der im Diagnosefall

i) bei mindestens einem der vorliegenden, zusammenhängenden Teilgraphen $G_i=(V_i,E_i)$ von G mit $v_i>t$ intakten Einheiten
ii) nach endlich vielen Zeitschritten eine korrekte Diagnose
iii) einschließlich der defekten Nachbarn von G_i bewirkt

d) für jede Einheit mindestens eine intakte Reserveeinheit existiert, die nicht in G enthalten ist und an den selben Knoten geschaltet werden kann

BEWEIS: notwendig:

a) Nach Annahme 9.1 ist ein gemeinsames Urteil der intakten Einheiten nötig. Dazu müssen sie einen zusammenhängenden Graphen, z.B. G', bilden. Um die defekten Einheiten auch tatsächlich abschalten zu können, muß $v'>t$ gelten.

b) Nehmen wir an, ein aufgetretener Defekt wird nicht erkannt oder eine Diagnose oder Reparatur nicht eingeleitet. Damit bleibt das System aber defekt.

c) Existiert kein Diagnosealgorithmus mit den genannten Eigenschaften, so können folgende Ereignisse eintreten:

i) Angenommen, es gibt nur einen zusammenhängenden Teilgraph G' mit $v'>t$ intakten Einheiten und der Diagnosealgorithmus bewirkt keine Diagnose, so gibt es keine Menge von t+1 Einheiten, die übereinstimmen und eine Reparatur veranlassen können. Also verbleibt das System defekt.

ii) Wenn die Diagnose unendlich viele Zeitschritte benötigt oder nicht korrekt ist, so kann es ebenfalls Einheiten geben, die defekt sind und nicht repariert werden.

iii) Angenommen, die defekten Nachbarn von V' werden nicht alle diagnostiziert. Dann kann dies der einzige Teilgraph mit mehr als t intakten, zusammenhängenden Einheiten sein, so daß die Defekten auch nicht von anderen Mengen von intakten Einheiten repariert werden und defekt bleiben.

d) Existiert für eine Einheit u_i keine intakte Reserveeinheit, so kann u_i nicht repariert werden und der Graph G von intakten Einheiten ist beim Ausfall von u_i nicht mehr gewährleistet.

hinreichend:

Angenommen, nicht mehr als t Einheiten sind ausgefallen und es gelte a),b),c) und d). Dann wird nach Auftreten eines Defekts mit b) eine Diagnose eingeleitet, es existieren mit a) mehr als t zusammenhängende, intakte Einheiten, die nach c) in endlich vielen Zeitschritten sich und die Defekte in ihrer Nachbarschaft korrekt diagnostizieren und mit Annahme 9.1 und d) auch korrekt reparieren. Damit ist die Zahl der intakten Einheiten um die der Nachbarschaft vermehrt. Mit endlich vielen Iterationen dieser Methode ist der gesamte Graph G korrekt diagnostiziert, repariert und enthält nach endlich vielen Zeitschritten nur intakte Einheiten, Q.E.D.

Ein Beispiel für eine solche Diagnose ist bei vollständigen Tests die iterative Bayesdiagnose in BAY3. Da bei dieser Diagnose erst die Testergebnisse verschickt und dann diagnostiziert wird, benötigt man für die Bayesdiagnose zusätzlich eine Formatcodierung (s. Kapitel 6) der Testergebnisse, um eine korrekte Absenderidentität zu garantieren. Die Bayesdiagnose ist auch bei der Teilmenge V''cV mit V'':=V' U D(V') dem größten, zusammenhängenden Teilgraph von intakten Einheiten G'=(V',E') und der defekten Nachbarschaft D(V') definiert (s. 8.2) und erneuert so im ersten Schritt alle defekten Nachbarn der intakten Einheiten von V'. Im zweiten Schritt wird auch D(D(V')) erneuert, so daß nach endlich vielen Schritten alle Einheiten getestet, diagnostiziert und eventuell erneuert worden sind. Nach Feststellung 5.3a ist das System damit fehlerfrei.

In der Definition für 't Fehler selbstreparierend' ist die Diagnose und Reparatur von Einheiten des Systems gefordert. Betrachten wir Rechnernetze, deren Einheiten dadurch repariert werden, daß ein Service-Techniker oder ein Platinenwechselmechanismus die defekten Baugruppen austauscht, so sind diese Rechnernetze nach der strengen Definition nicht selbstreparierend. Trotzdem ist es sinnvoll, daß auch diese Rechnernetze zumindest die Diagnose und den Reparaturauftrag selbst bestimmen, da der Techniker oder der Wechselmechanismus in großen Netzen damit überlastet wäre.
Betrachten wir den Reparaturmechanismus als ausfallsicher und als Teil des Systems und gilt Annahme 9.1 anstelle der eigentlichen Reparatur für einen Reparaturauftrag, so ist auch für ein solches System Satz 9.1 gültig.

Im Satz 9.1 wird davon ausgegangen, daß für jede Einheit mindestens eine redundante Einheit mit einem lokalen Umschaltmechanismus im Fehlerfall zur Verfügung steht. Eine derartige Auslegung des Systems ist natürlich nur sinnvoll, wenn die Zuverlässigkeit des Systems dadurch erhöht wird. Betrachten wir dies näher.

Sei ein System S_1 gegeben, das mit N Einheiten den Graphen G realisiert. Für jede der N Einheiten existiere eine Reserveeinheit. Die Umschaltung auf die Reserveeinheit soll mit dem oben beschriebenen, lokalen Mechanismus geschehen. Betrachten wir nun ein zweites System S_2, das ebenfalls 2N Einheiten aufweist, aber keinen solchen lokalen Umschaltmechanismus besitzt. Stattdessen sind die N Reserveeinheiten in einem zweiten Netz enthalten, das ebenfalls G realisiert und im Fehlerfall an die Stelle des ersten geschaltet wird. Welches System ist zuverlässiger?

Wenn wir den Umschaltmechanismus als Ultrazuverlässig betrachten (geringe Zahl von Transistorfunktionen) und beide Systeme mit Hilfe von Z-Netzen/STö/ modellieren, so erhalten wir die Z-Netze in Abb.9.1a.

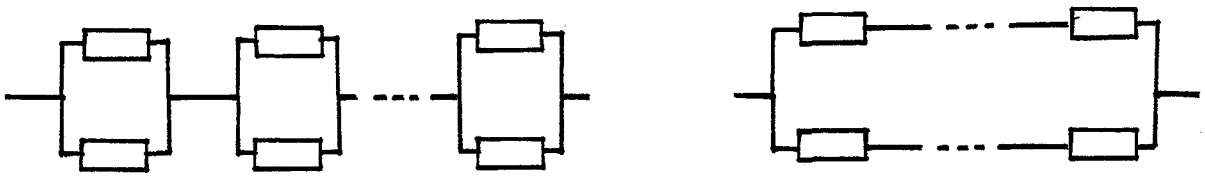


Abb. 9.1a Z-Netze der Systeme S_1 und S_2

Mit der für alle Einheiten gleichen Zuverlässigkeit R ist die Zuverlässigkeit für S_1 $P_1(R,N)=(2R-RR)^N = R^N(2-R)^N$ und
für S_2 ist $P_2(R,N)=2R^N-R^N\ R^N = R^N(2-R^N)$.
Für N=1 sind beide Z-Netze gleich, also $P_1(R,1)=R(2-R)=P_2(R,1)$.
Bei R=1 kann keine Einheit ausfallen, also gilt auch hier die Gleichheit $P_1(1,N)= 1 =P_2(1,N)$.
Bei R=0 sind immer alle Einheiten ausgefallen, also ist $P_1(0,N)= 0 =P_2(0,N)$.

BEHAUPTUNG: Für alle $N>1$, $0<R<1$ ist das System S_1 zuverlässiger als das System S_2.

BEWEIS: Durch vollständige Induktion.

Induktionsanfang:

Sei $N=2$.

Mit $R\neq 1$ ist $(R-1)^2 > 0$. /*2, +2

Also ist $4-4R+2R^2 > 2$

und $(2-R)^2 > 2-R^2$ / $*R^2$

und somit $P_1(R,2) > P_2(R,2)$.

Induktionsdurchführung:

Sei die Relation für $N=n\geq 2$ bewiesen (Induktionsannahme).

Mit $R < 1$ ist $R^n < 1$ /*-(1-R)

und so $2R^n(R-1) > 2(R-1)$ /+4, -2R, $-R^{n+1}$

$4-2R^n-2R+R^{n+1} > 2-R^{n+1}$

$(2-R^n)(2-R) > 2-R^{n+1}$

Also ist mit $(2-R)^n > 2-R^n$ (Induktionsannahme)

auch $(2-R)^{n+1}=(2-R)^n(2-R)>(2-R^n)(2-R)_> 2-R^{n+1}$

und somit $P_1(R,n+1) > P_2(R,n+1)$.

Also gilt die Relation für $N=n+1$ und damit

für alle $N\geq 2$ und $0<R<1$, Q.E.D.

Also ist das System S_1 im allgemeinen zuverlässiger als S_2.

Beispiel:

Bei $N=100$ Einheiten und $R=0.99$ funktioniert das System S_1 mit der Wahrscheinlichkeit $P_1=0.99$, das andere System S_2 dagegen mit der Wahrscheinlichkeit $P_2=0.59$.

Bemerkenswert am Satz 9.1 ist die Tatsache, daß an die Stelle einer bei jedem Schritt garantiert zu lokalisierenden, defekten Einheit (z.B. in der sequentiellen Diagnose aus 2.1) die Forderung getreten ist, daß bei t defekten Einheiten immer eine für die iterative Diagnose nötige, zusammenhängende, intakte Menge von mehr als t Einheiten vorhanden sein muß.

Damit stellt sich die Frage:

> Wieviel Einheiten dürfen in einem Rechnernetz mit gegebener Struktur ausfallen, um unter diesen Nebenbedingungen immer noch ein intaktes, sich selbst reparierendes System zu gewährleisten?

Was ist also die maximale Größe von t, um bei N Einheiten, gegebenen G, Zusammenhangssgrad k(G) und beliebiger Anordnung von $r \leq t$ defekten Einheiten immer ein zusammenhängendes Subsystem von mehr als t intakten Einheiten zu garantieren, das durch gemeinsames Votum defekte Einheiten abschalten kann?
Diese Frage wird im nächsten Abschnitt am Beispiel der regulären, geschlossenen Flächennetze näher untersucht.

9.2 Selbstreparatur in regulären Flächennetzen

Eine korrekte Diagnose und Reparatur der VLSI-Chips ist nicht nur im Normalbetrieb, sondern auch für die Herstellung der Chips nötig. Da die Ausbeute an vollständig funktionierenden Chips bei der Höchstintegration meist nicht sehr hoch ist, wird bei der Produktion dazu übergegangen, redundante Chip-Funktionen (z.B. Speicherelemente bei Speicherchips) vorzusehen, die durch Schmelzschalter die Funktionen defekter Chipteile übernehmen können.

Diese Schalter (Diodenstrecken, die durch Stromüberlastung nichtleitend werden, oder Metallbrücken, die mit Laserstrahlen durchtrennt werden) funktionieren aber nur einmal.

Deshalb sind die im vorigen Abschnitt besprochenen Fragen in VLSI-Rechnernetzen besonders interessant. VLSI-Chips erlauben i.A. zwar eine Rekonfiguration des Systems mit Reserveeinheiten bei der Herstellung oder beim Start-up Test, gestatten aber durch die begrenzte Zahl von Anschlußpunkten von außen keinen direkten Zugang zu jeder einzelnen Einheit, so daß ein auf dem Chip ablaufendes Testverfahren wünschenswert erscheint. Außerdem wird die Testzeit und damit die Testkosten der Chips durch die parallel auf dem ganzen Netz ablaufenden Tests stark verkleinert. (vgl. auch /AMM/).

Es soll nun für die in 7.1 eingeführten, rückseitig geschlossen Flächennetze die Frage näher untersucht werden:

> Wie klein kann in diesen Systemen bei r Defekten die größte, zusammenhängende Menge von intakten Einheiten werden?

Betrachten wir die damit eng verknüpfte Frage:

> Wieviel Einheiten lassen sich maximal durch r Defekte von der Gesamtmenge aller intakten Einheiten isolieren?

Dabei sei 'isoliert' folgendermaßen definiert:

DEFINITION:

> Eine Menge V_1 von Einheiten in einem Graphen G heiße 'blockiert', falls gilt
>
> 1) der Teilgraph $G_1=(V_1,E_1)$ von G ist zusammenhängend und enthält nur intakte Einheiten
>
> 2) alle Nachbareinheiten $D(V_1)$ von V_1 sind defekt (Elemente von G_0)
>
> Die Menge V_1 heiße 'isoliert', wenn außerdem noch gilt
>
> 3) $v-v_1-v_0 \geqq v_1$

Eine isolierte Menge von Einheiten hat also immer weniger oder gleich viele, ausschließlich intakte Einheiten wie die restliche Menge von intakten Einheiten, von der sie isoliert ist.

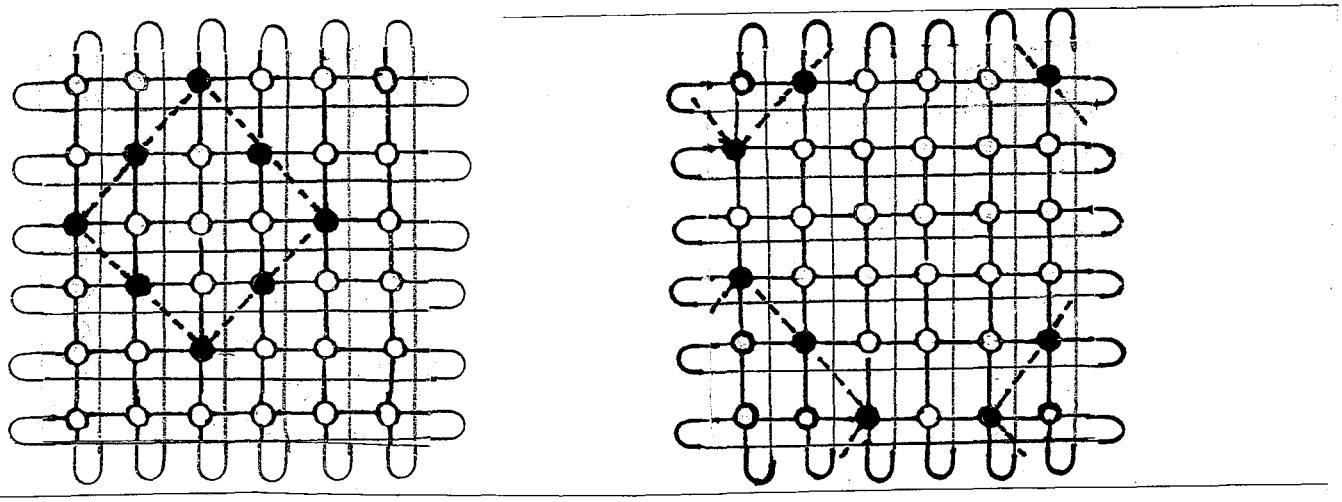
Die Blockierung und Isolation eines Teilgraphen läßt sich analog zu oben

definieren:
Ein Teilgraph heißt 'blockiert' ('isoliert'), wenn seine Knotenmenge V_1 blockiert (isoliert) ist.
Angenommen, wir identifizieren die Einheiten mit Punkten in einer Ebene. Dann ist bekanntermaßen der Kreis die Figur, die mit gegebenem Umfang (Punkten auf einer Linie) den größten Flächeninhalt (Zahl der zusammenhängenden Punkte) abtrennen (isolieren) kann. Es liegt die Vermutung nahe, daß ähnliches auch für die diskrete Topologie der Flächennetze gilt.
DEFINITION:

Ein Teilgraph $G_1 \subset G$ heiße 'r-maximal' für $r \in \mathbb{N}$, falls gilt
a) es gibt eine Menge von genau r defekten Einheiten, die G_1 isolieren
b) G_1 ist maximal, d.h. es existiert kein anderer Teilgraph, der a) erfüllt und eine größere Zahl von Einheiten besitzt.

Bemerkung: Der r-maximale Teilgraph ist i.A. nicht eindeutig.

Als Beispiel einer Isolation betrachten wir Abb.9.2a. Bei einem 'ausreichend großen', geschlossenen Flächennetz hängt ein isolierter Teilgraph nicht vom Rand der Flächennetze ab. In Abb.9.2b ist der gleiche isolierte Teilgraph aus Abb.9.2a zu sehen, da die Nachbarschaftsrelationen der isolierten Einheiten die selben sind.

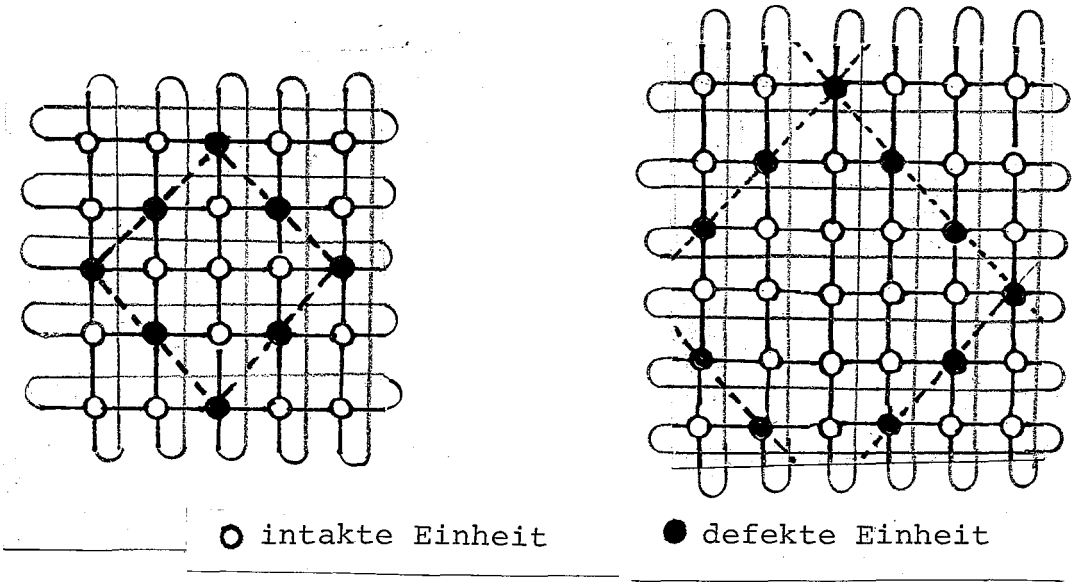


a) b)

Abb. 9.2a,b Beispiele isolierter Teilgraphen

Was bedeutet nun 'ausreichend groß'?
Betrachten wir die folgenden Abbildungen. In Abbildung 9.2c ist ein rückseitig geschlossenes Netz mit N=25 und r=8 Defekten eingezeichnet, die maximal $v_1=5$ intakte Einheiten von $v'=12$ anderen isolieren. Sind stattdessen r=10 Einheiten

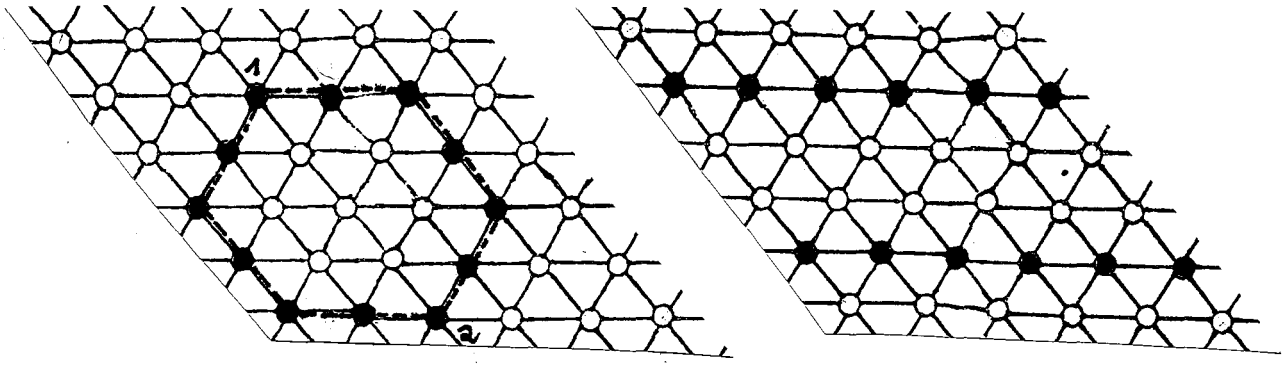
defekt, so werden in Abb.9.2d $v_1=13=v'>r$ intakte Einheiten voneinander isoliert.



c) r=8, N=25 d) r=10, N=36

Abb.9.2c,d Randeffekte bei Flächennetzen

In Abb.9.2e,f sind für N=36, r=12 zwei isolierte Teilgraphen gezeigt. Die Schließung der Netze ist nicht eingezeichnet (s.7.1).



e) f)

Abb. 9.2e,f Randeffekte bei Flächennetzen

Der in Abb.9.2e abgebildete Teilgraph ist in 'ausreichend großen' Flächennetzen nach Lemma 9.2b (s. unten) r-maximal. In den geschlossenen Flächennetzen läßt sich mit den vorhandenen Defekten statt des r-maximalen Teilgraphen mit $v(r)=7$ isolierten Einheiten auch ein Teilgraph mit $v_1=12=v'=r$ Einheiten (Abb.9.2f) isolieren.

Die Ursache für die größere Zahl von isolierten Einheiten in Abb.9.2e gegenüber Abb.9.2f liegt zweifelsohne in der Tatsache begründet, daß die Flächennetze nicht unendliche Ausdehnung haben, sondern geschlossene Ränder besitzen.

DEFINITION:

Wenn bei r defekten Einheiten sich in endlichen, geschlossenen Flächennetzen mehr intakte Einheiten isolieren lassen als in einem Flächennetz unendlicher Ausdehnung, so liegt ein 'Randeffekt' vor.

Die Vorraussetzung für Lemma 9.2b ist, daß bei den betrachteten, geschlossenen Flächennetzen keine Randeffekte auftreten.

Wann tritt nun ein Randeffekt auf?

In einem Flächennetz unendlicher Ausdehnung können die zusammenhängenden, intakten Einheiten von G_1 auf der selben Gitterlinie nur dann von den anderen Einheiten aus G isoliert werden, wenn auf der Gitterlinie mindestens zwei Defekte existieren. Ist die Gitterlinie aber in sich geschlossen und hat damit eine endliche Zahl von intakten Einheiten, so können auch nur Einheiten von G_1 und damit auch nur ein oder kein Defekt auf einer Gitterlinie vorhanden sein, ohne daß die Isolation damit aufgehoben wäre.

Der Randeffekt besteht also aus der Tatsache, daß bei einer Gitterlinie der betrachteten, geschlossenen Flächennetze unter Umständen weniger als zwei Defekte zur Isolierung der intakten Einheiten der Gitterlinie nötig sind und damit bei gleicher Zahl von Defekten mehr intakte Einheiten in einem Teilgraphen isoliert werden können als in einem Flächennetz unendlicher Ausdehnung.

Feststellung 9.2:

In einem geschlossenen Flächennetz ist ein Randeffekt d.u.n.d vorhanden, wenn es bei r Defekten einen isolierten Teilgraphen G_1 gibt und mindestens eine den Teilgraphen kreuzende Gitterlinie weniger als zwei zur Isolierung nötigen Defekte aufweist.

ANNAHME 9.2:

Im Folgenden betrachten wir nur Flächennetze, die gemäß Kapitel 7.1 geschlossenen sind mit $N=n^2, n\in\mathbb{N}$ Einheiten und $k(G)=4$ oder 6 aufweisen.

Da bei $k(G)=3$ eine Definition von 'Gitterlinien' ungleich schwieriger ist als bei $k(G)=4$ und 6, wird dieses Flächennetz im Folgenden nicht betrachtet.

Für den Satz über hinreichende Bedingungen für t-Fehler Selbstreparatur in regulären Flächennetzen müssen zuerst die folgenden drei Hilfssätze einführt werden, deren Beweis aus Gründen der Übersicht erst am Schluß dieses Abschnittes erfolgt.

LEMMA 9.2a:

Bei Flächennetzen nach Annahme 9.2 tritt für beliebiges n und bei beliebiger Lage von r Defekten kein Randeffekt d.u.n.d. auf, wenn $r<2n-2$ ist.

BEWEIS: siehe unten.

Sei der Abstand zwischen zwei Einheiten in G als die Kantenzahl des Weges zwischen den beiden Einheiten, der die kleinste Kantenzahl hat, definiert. Dann gilt das folgende Lemma:

LEMMA 9.2b:

Es existieren r Defekte in einem Flächennetz und es gelte Annahme 9.2 sowie $N \geq (r/2+1)^2$.

Dann werden maximal $v(r)=\lfloor 1+r/2(r/k(G)-1) \rfloor$ intakte Einheiten in einem Teilgraphen isoliert. Bei $r \leq k(G)$ existiert für gerades r ein Punkt, von dem alle Defekte den gleichen Abstand haben.

BEWEIS: siehe unten.

LEMMA 9.2c:

Gelten die Vorraussetzungen von Lemma 9.2b, so werden mit r Defekten in den regulären Flächennetzen mit $k(G)=4,6$ in einem einzigen r-maximalen Teilgraphen durch insgesamt r defekte Einheiten immer mindestens so viele intakte Einheiten isoliert wie in einer Vereinigung von zwei oder mehr isolierten Teilgraphen.

Damit gilt

SATZ 9.2:

Die Flächennetze nach Annahme 9.2 sind t-Fehler selbstreparierend, wenn bei N Einheiten im System maximal $t:=2\sqrt{N}-3$ Einheiten defekt sind, $N \geq 4$ ist und mit Annahme 9.1 die Bedingungen b),c) und d) aus Satz 9.1 vorliegen.

BEWEIS:

Mit $t=2\sqrt{N}-3$ ist die Voraussetzung für die Lemmata 9.2b und 9.2c erfüllt. Also wird durch $r \leq t$ Defekte die größte Zahl von intakten Einheiten in einem einzigen, r-maximalen Teilgraphen G_1 von G' isoliert und G_1 hat $v_1=\lfloor 1+(r/2)(r/k(G)-1) \rfloor$ intakte Einheiten. Damit ist, unabhängig von der Lage der Defekte, ein zusammenhängender Graph $G'=G-G_1-G_0$ mit $v'=N-v_1-r$ intakten Einheiten vorhanden.

Wann ist v' größer als die Zahl der Defekte?

Sei in der folgenden Rechnung k(G) mit k abgekürzt.

Da

$$N \geqq 4 \qquad \text{ist } t \geqq 1 > 0.$$

Mit

$$\frac{k-2}{4k} > 0$$

ist

$$t^2\left(\frac{k-2}{4k}\right) + \frac{5}{4} > 0$$

$$t^2\left(\frac{1}{4} - \frac{1}{2k}\right) + \frac{6}{4}t + \frac{9}{4} - 1 + \frac{t}{2} - t > t$$

$$\left(\frac{t+3}{2}\right)^2 - \left(1 + \frac{t}{2}\left(\frac{t}{k} - 1\right)\right) - t > t$$

Mit

$$N = \left(\frac{t+3}{2}\right)^2 \text{ und } v(t) = 1 + \frac{t}{2}\left(\frac{t}{k} - 1\right)$$

ist

$$N - v(t) - t = v' > t \geqq r$$

Somit ist für G' auch Bedingung a) von Satz 9.1 erfüllt und Satz 9.2 damit bewiesen.

Bemerkung:

Werden Flächennetze verwendet, die auf verschiedenen Gitterlinien einen unterschiedlichen Umfang haben (z.B. 'rechteckige' Netze), so sind die Betrachtungen jeweils für ein quadratisches Teilstück mit der Seitenlänge des kleinsten Umfangs gültig. Sind die Bedingungen der Selbstreparatur für ein solches Teilnetz erfüllt, so sind sie auch für das gesamte Flächennetz hinreichend.

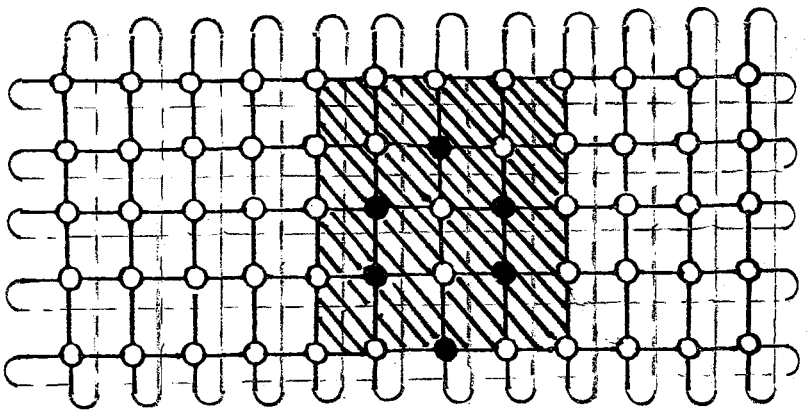


Abb. Quadratisches Teilstück (schraffiert) eines 'rechteckigen' Flächennetzes

9.2.1 BEWEIS von Lemma 9.2a

notwendig:

Ist $r \geq 2n-2$, so gibt es isolierte Teilgraphen mit Randeffekten. Beispiele dafür sind die Konfigurationen bei $k(G)=4$, $n=6$, $r=2n-2=10$ Defekten in Abb.9.2d und bei $k(G)=6$, $n=6$, $r>2n-2$ in Abb.9.2e,f.

hinreichend:

Um zu zeigen, daß die Bedingung $r<2n-2$ hinreichend für die Abwesenheit von Randeffekten in isolierten Teilgraphen ist, reicht es mit Feststellung 9.2 aus, zu zeigen, daß unter dieser Bedingung die intakten Einheiten einer Gitterlinie, auf der weniger als zwei Defekte liegen, nicht Einheiten von V_1 eines isolierten Teilgraphen G_1 sein können.

I) $k(G)=4$

Behauptung: Ist $r=2n-3$, so existiert kein Teilgraph G_1, der isoliert ist und

a) eine Gitterlinie ohne einen zur Isolation nötigen Defekt

oder

b) eine Gitterlinie mit nur einem zur Isolation nötigen Defekt

besitzt. Also existiert nach Feststellung 9.2 kein isolierter Teilgraph mit einem Randeffekt.

Beweis: Es existiere ein mit $r=2n-3$ blockierter Teilgraph G_1, der einen Randeffekt aufweist. Dann ist nach Feststellung 9.2 eine der beiden Fälle a) oder b) gegeben.

a) In G_1 existiert mindestens eine Gitterlinie g_a, die keinen zur Isolation nötigen Defekt und somit nur intakte Einheiten aus G_1 enthält.
Da ein Teilgraph nicht von der Lage im Netz abhängig ist, sei o.B.d.A. die Gitterlinie g_a am oberen Rand der Abb. 9.2.1a horizontal angeordnet. Die Schließung des Flächennetzes ist nicht eingezeichnet.

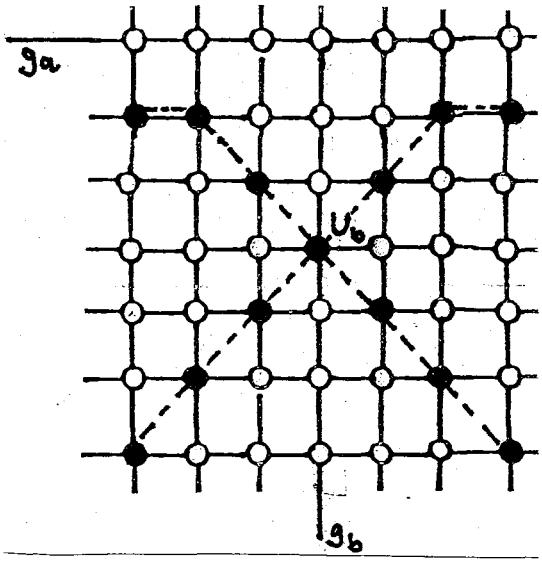


Abb. 9.2.1a Blockierung eines Teilgraphen mit n=7

Es können nicht alle n Einheiten von g_a vertikal mit je zwei Defekten blockiert sein, sondern es muß mindestens eine vertikale Gitterlinie ohne Defekt oder mit nur einem Defekt existieren. Im Folgenden wird gezeigt, daß bereits bei nur einer vertikalen Gitterlinie g_b mit einem Defekt u_b kein Teilgraph G_1 existiert, der g_a und g_b enthält und isoliert ist. Da eine vertikale Gitterlinie ohne Defekt oder mehrere mit je einem Defekt mehr Intakte enthalten als g_b, gibt es somit keinen Teilgraphen, der g_a enthält und mit 2n-3 Defekten isoliert ist.
Um die intakten Einheiten sowohl vertikal von g_a als auch horizontal von g_b zu blockieren, müssen auf jeder vertikalen Gitterlinie außer auf g_b und auf jeder horizontalen Gitterlinie außer auf g_a notwendigerweise je zwei Defekte existieren. Um den Teilgraphen auch hinreichend zu blockieren, müssen die Defekte dabei horizontal und vertikal versetzt angeordnet sein.
Zwei Fälle werden unterschieden: n ungerade und n gerade.

i) n ungerade
In Abb.9.2.1a ist die Situation bei n=7 zu sehen. Mit der Nebenbedingung der versetzten Einheiten ist u_b in der Mitte der Abbildung angeordnet. Da die versetzten Defekte fast Diagonale über das Flächennetz bilden, sind mit den Einheiten von g_a mehr intakte Einheiten in V_1 als in der Restmenge, so daß G_1 nicht isoliert sein kann.

ii) n gerade
Bei geradem n läßt sich g_b so anordnen, daß zwischen ihr und einer

Nachbargitterlinie die vertikale Mittellinie des Netzes verläuft (s. Abb 9.2.1b).

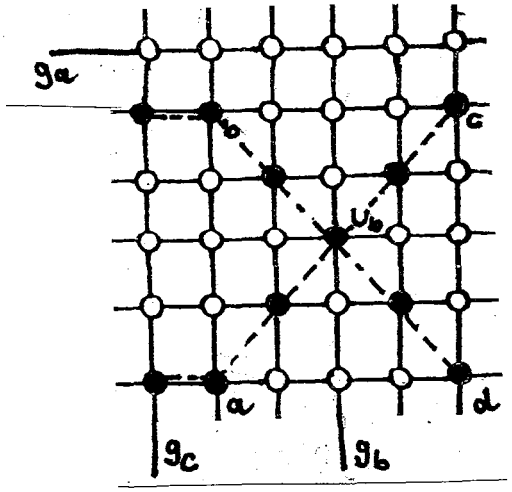


Abb 9.2.1b Ein blockierter Teilgraph mit n=6

Der Defekt u_b liegt in der Mitte der um die Einheit von g_a reduzierten Gitterlinie g_b. In dem Teilstück des Flächennetzes, das die Einheiten a,b,c,d als Eckpunkte hat, ist die Zahl der zu V_1 gehörenden Einheiten gleich der Zahl der zur Restmenge V' zählenden Einheiten. Da auf g_a mehr intakte Einheiten existieren als auf g_c, ist $v_1 > v'$ und G_1 somit nicht isoliert.

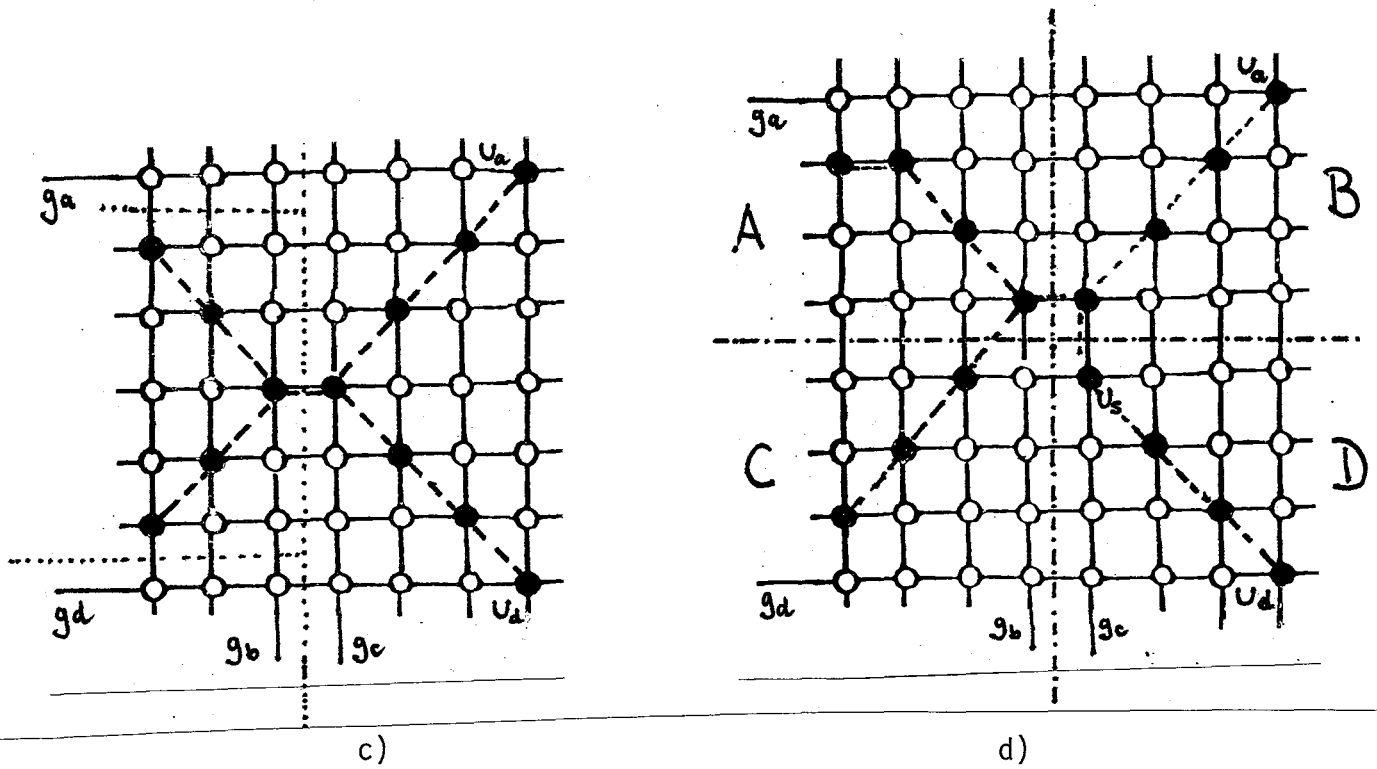
b) Es gibt mindestens eine Gitterlinie g_a, die nur einen zur Blockierung nötigen Defekt enthält und damit intakte Einheiten ausschließlich aus G_1. Um die n-1 intakten Einheiten von g_a vertikal ohne Randeffekt zu isolieren, sind mindestens 2n-2 Defekte nötig. Da nur 2n-4 Defekte zur vertikalen Blockierung vorhanden sind, müssen mindestens zwei vertikale Gitterlinien mit je einem zur Isolation nötigen Defekt oder mindestens eine Gitterlinie ohne Defekt vorhanden sein. Da in a) bereits bewiesen wurde, daß es keinen isolierten Teilgraph mit einer Gitterlinie mit einem Defekt und einer dazu orthogonalen Gitterlinie ohne Defekt gibt, bleibt nur noch zu beweisen, daß es auch keinen isolierten Teilgraphen mit einer Gitterlinie g_a mit einem Defekt und zwei dazu orthogonalen Gitterlinien g_b,g_c mit je einem Defekt gibt.
Angenommen, es gibt einen blockierten Teilgraphen G_1, der g_a, g_b und g_c enthält. G_1 muß noch eine zweite horizontale Gitterlinie g_d mit einem Defekt enthalten, da
- mit r-1=2n-4 Defekten nicht n-1 horizontale Gitterlinien mit je zwei Defekten besetzt werden können (im Gegensatz zu Abb.9.2.1b, wo u_b auf einer Gitterlinie ausreicht)
- eine zweite horizontale Gitterlinie ohne einen zur Blockierung nötigen

Defekt kann mit einer vertikalen Gitterlinie und einem Defekt nach Beweisschritt a) nicht einem isolierten Teilgraphen angehören.

Sei die zweite horizontale Gitterlinie g_d o.B.d.A. Nachbargitterlinie von g_a. Die beiden Defekte u_a und u_b befinden sich auf der gleichen vertikalen Gitterlinie, da sonst zusätzliche Defekte zur vertikalen Blockierung auf den Gitterlinien von u_a und u_b sein müssen und die Zahl der Defekte für eine Blockierung ungenügend ist.

i) n ungerade

Da obiges Argument auch für g_b und g_c gilt, ergibt sich die Situation in Abb.9.2.1c.



c) d)

Abb 9.2.1 c,d Blockierung bei Randeffekten

Aus der vorhandenen Geometrie ist zu ersehen, daß in V_1 mehr intakte Einheiten sich befinden als in der Restmenge. Also gibt es keinen isolierten Teilgraphen, der g_a, g_b, g_c und g_d enthält.

ii) n gerade

Wenn wir wieder annehmen, daß g_a und g_d und die Defekte u_a und u_d benachbart sind, so ist eine für die Blockierung nötige Versetzung der Defekte bei der geraden Zahl von Gitterlinien nicht möglich (s.Abb.9.2.1d). Fügen wir nun einen weiteren Defekt u_s dem Flächennetz zu, so ergibt sich bei einer Einteilung des Netzes in die Teilstücke A,B,C,D, daß in B und D die Anzahl der intakten Einheiten aus V_1 und aus V' gleich sind, dagegen in A und C offensichtlich mehr Einheiten zu V_1 gehören. Also gibt es für

diese Situation keinen isolierten Teilgraphen, der g_a, g_b, g_c und g_d enthält. Sind in G_1 anstelle von u_s weitere Gitterlinien mit Randeffekt enthalten, so erhöht sich v_1, so daß der so blockierte Teilgraph ebenfalls nicht isoliert ist, Q.E.D.

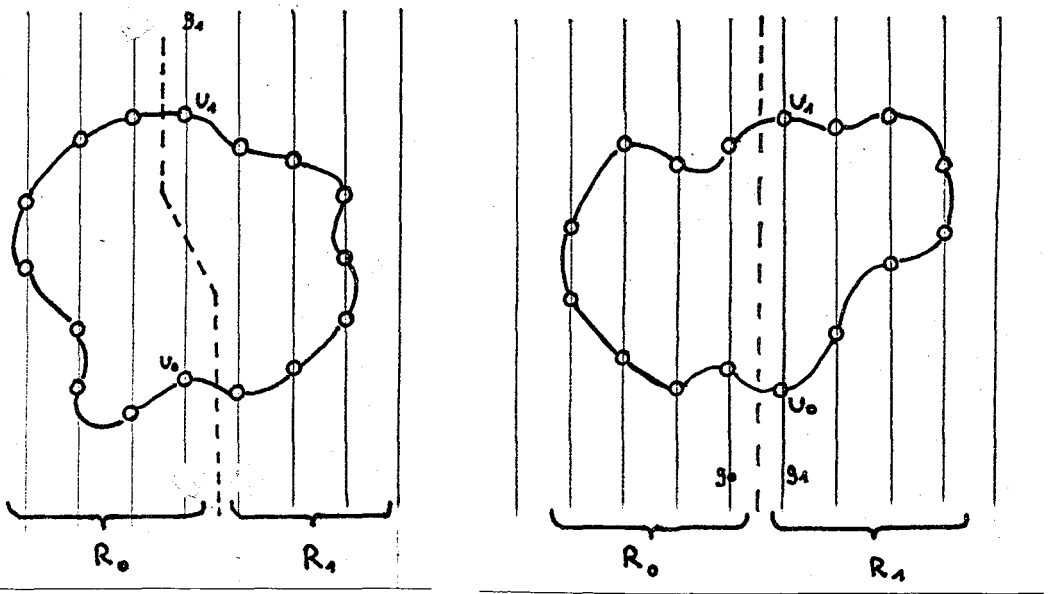
In dem vorhergehenden Beweis wurde für $r=2n-3$ bewiesen, daß es keinen isolierten Teilgraphen mit Randeffekt gibt. Nun ist eine Blockierung eines Teilgraphen mit $r<2n-3$ Defekten nur dann möglich, wenn es mindestens eine weitere Gitterlinie mit Randeffekt gibt. In den Beweisteilen a) und b) ist somit mindestens eine zusätzliche horizontale oder vertikale Gitterlinie mit intakten Einheiten in G_1 vorhanden. Dies bedeutet, daß der mit weniger als 2n-3 Defekten blockierte Teilgraph eine größere Anzahl oder gleich viele Einheiten besitzt wie der mit 2n-3 Defekten blockierte Teilgraph. Also kann auch kein mit $r<2n-3$ Defekten blockierter Teilgraph isoliert sein.

II) $k(G)=6$

Bei gleicher Zahl von Defekten sind zur Blockierung von intakten Einheiten bei $k(G)=6$ durch den größeren Zusammenhang gleichviel oder mehr Gitterlinien mit Randeffekt als bei $k(G)=4$ nötig. Also ist auch bei $k(G)=6$ ein mit $r<2n-2$ Defekten blockierter Teilgraph nicht isoliert, Q.E.D.

9.2.2 BEWEIS von Lemma 9.2b

Betrachten wir die geschlossenen, reguläre Flächennetze aus Annahme 9.2.
Bei $r<k(G)$ können keine intakten Einheiten isoliert werden. Mit $v(r)=0$ ist die Formel aber trotzdem korrekt.
Sei im Folgenden $r \geq k(G)$.
Sei eine 'Gitterlinie' definiert als ein geschlossener Weg von Kanten einer Orientierung, beispielsweise horizontal oder vertikal. Zwei Gitterlinien sind benachbart, wenn es für jeden Knoten einer Gitterlinie einen benachbarten Knoten der anderen Gitterlinie gibt.
Seien die regulären Flächennetze mit $k(G)=4$ und 6 so ausgerichtet, daß es vertikale Gitterlinien gibt.
Der r-maximale Teilgraph hat zweifelsohne die Eigenschaft, daß die Reparatur einer beliebigen defekten Einheit ('Randpunkt') ausreicht, bei einem solchen Teilgraphen die Isolation aufzuheben.
Angenommen, es gibt einen r-maximalen Teilgraphen. Betrachten wir seine Randpunkte auf den vertikalen Gitterlinien in Abb.9.2.2a näher.



a) asymmetrische Teilung b) symmetrische Teilung

Abb.9.2.2a,b Teilungen der Randpunkte des r-maximalen Teilgraphen

Seien die Randpunkte von G_1 in zwei Mengen R_0 und R_1 unterteilt. Mit $r_0:=|R_0|$ und $r_1:=|R_1|$ ist definitionsgemäß $r=r_1+r_2$.

A) ASYMMETRISCHE TEILUNG

Betrachten wir eine Gitterlinie g_1 mit r_g Defekten. Auf den Gitterlinien links von g_1 seien r_0', auf den Gitterlinien rechts von g_1 r_1' Defekte. Dabei gelten folgende Relationen: $r_1' + r_g > r_0'$, $r_0' + r_g > r_1'$

Die Menge V_0 der Defekten läßt sich also nur asymmetrisch aufteilen:

$R_0 := \{u_i / u_i \varepsilon V_0$ und in Abb.9.2.2a 'links' von $g_1\} \cup \{u_0\}$.

$R_1 := \{u_i / u_i \varepsilon V_0$ und in Abb.9.2.2a 'rechts' von $g_1\} \cup \{u_1\}$.

Sind noch weitere Defekte auf g_1, so sollen sie ebenfalls möglichst gleichmäßig aufgeteilt werden.

B) SYMMETRISCHE TEILUNG:

Seien zwei Gitterlinien g_0 und g_1 so ausgewählt, daß sie den r-maximalen Teilgraphen G_1 kreuzen und benachbart sind.

$R_0 := \{u_i / u_i \varepsilon V_0$ und auf g_0 oder einer Gitterlinie in Abb.9.2.2 b 'links' von $g_0\}$ und $R_1 := V_0 - R_0$.

Lassen sich g_0 und g_1 so auswählen, daß $r_0 = r_1$ ist, dann ist r gerade.

Es ist zu zeigen, daß bei r Defekten maximal $v(r) = \lfloor 1 + r/2(r/k-1) \rfloor$ intakte Einheiten in G_1 isoliert werden können.

Sei r aufteilbar in $r = r_k + n$, wobei r_k ohne Rest durch k(G) teilbar und für die Variable n gilt $0 \leq n < k(G)$. Dann ist obige Behauptung bei $n > 0$ äquivalent zu der Behauptung

$$v(r_k+n) = \left\lfloor v(r_k) + n r_k/k(G) + \frac{n(n-k(G))}{2k(G)} \right\rfloor = v(r_k) + n r_k/k(G) - 1,$$

da mit $n > 0$, $n < k = 4,6$ $\quad -\frac{n(n-k(G))}{2k(G)} < 0$ und größer als -1 ist.

I) Betrachten wir nun das reguläre Flächennetz mit $k(G) = 4$ in Abb.9.2.2c, das 'ausreichend' groß sein soll.

A) ASYMMETRISCHE TEILUNG

(1) Sei r gerade und $r_0 = r_1$, also $r = r_k + 2$

Sei zwischen u_0 und u_1 der kürzeste Weg ID mit dem Abstand (Kantenzahl) id.

Dann ist $r_1 = id$, weil

a) Sei $r_1 < id$.

Zwischen u_0 und u_1 gibt es id-1 Knoten (Einheiten), durch die jeweils eine horizontale Gitterlinie läuft.

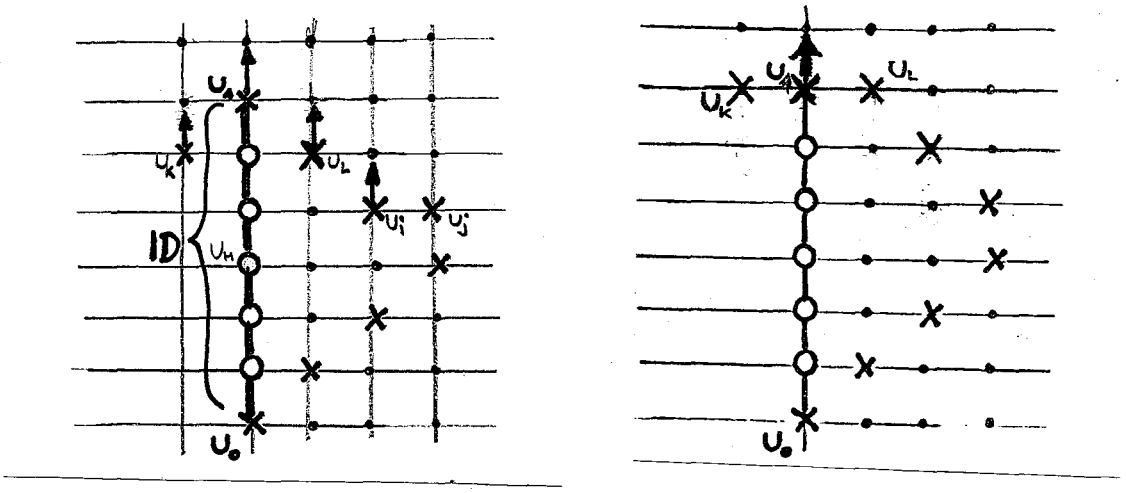


Abb.9.2.2c,d Konfiguration der Menge R_1

Auf jeder Gitterlinie muß mindestens ein Defekt aus R_1 zu finden sein, da der Teilgraph G_1 sonst nicht isoliert wäre. Inklusive u_1 und u_2 sind damit mindestens 2id Defekte zur Isolation von G_1 nötig. Ist dagegen $r_1<id$, so ist mit $r=r_1+r_2<2id$ G_1 im Widerspruch zur Voraussetzung nicht isoliert.

b) Sei $r_1>id$

Sei ein 'Nachbardefekt' eines Randpunkts u_i ein Defekt, der auf einer benachbarten oder gleichen horizontalen und vertikalen Gitterlinie existiert. Ein zur Isolation nötiger Defekt hat also mindestens zwei Nachbardefekte.

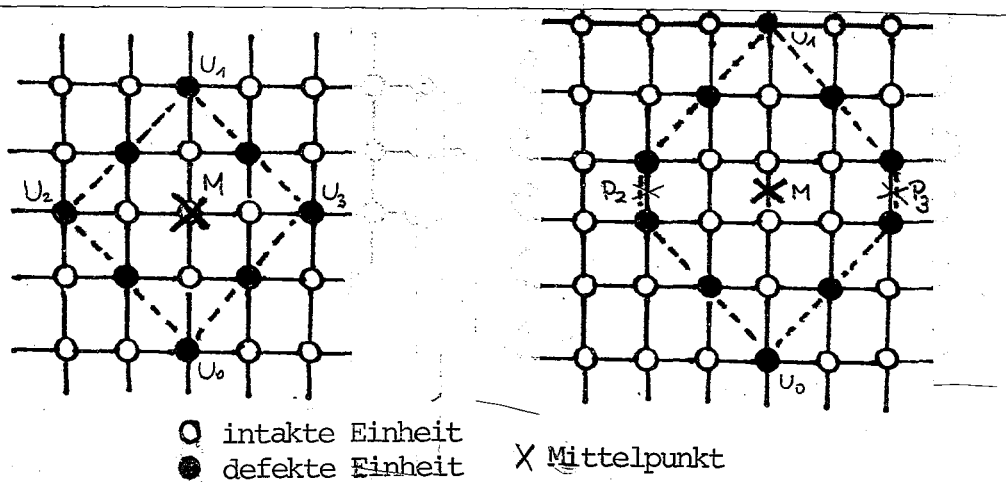
Dann gibt es auf einer horizontalen Gitterlinie zwei Defekte u_i und u_j (s.Abb.9.2.2c), wobei die Nachbardefekte von u_i auf der selben oder der benachbarten, höheren (tieferen) horizontalen Gitterlinie liegen.

i) Angenommen, u_i und u_j sind auf den horizontalen Gitterlinien, die zwischen u_1 und u_0 ID kreuzen, d.h. Einheiten mit ID gemeinsam haben. Dann gibt es jeweils eine Konfiguration, in der anstelle von u_i ihr vertikaler Nachbar (In Abb.9.2.2d die Einheit über u_i) defekt ist und der so entstandene Graph G_1' trotzdem isoliert ist. Da G_1' eine Einheit zusätzlich enthält, war G_1 nicht r-maximaler Teilgraph, im Widerspruch zur Voraussetzung.

ii) Angenommen, u_i und u_j sind auf horizontalen Gitterlinien, die ID nicht kreuzen; d.h. auf der selben oder einer benachbarten Gitterlinie von u_0 bzw. u_1(beispielsweise die Einheiten u_k,u_1,u_1 in Abb.9.2.2d). Aus

Symmetriegründen (Spiegelsymmetrie der Netzstruktur um ID und Rotationssymmetrie um 90 Grad) existieren dann mit den Einheiten aus R_0 mindestens drei Defekte auf einer horizontalen Gitterlinie und es gibt eine Konfiguration, bei der statt u_0 der untere bzw statt u_1 der obere Nachbar defekt ist. Bei dieser Konfiguration werden bei gleichem r mehr Intakte isoliert, so daß der Teilgraph nicht r-maximal war, im Widerspruch zur Voraussetzung.

Mit der Argumentation von i) und ii) müssen die Defekte auf einer Hälfte verschiedenen, einander benachbarten, horizontalen Gitterlinien sein. Da die Netzgeometrie nach einer Rotation um 90 Grad die gleiche ist, müssen die Defekte auch auf einander benachbarten, vertikalen Gitterlinien sein. Betrachten wir u_0 in Abb 9.2.2d. Nach den bisherigen Ausführungen gibt es nur auf den horizontalen Gitterlinien, die ID kreuzen, defekte Einheiten. Da die Defekte aus den Symmetriegründen sich auch auf den vertikalen Gitterlinien parallel zu ID befinden, sind die dem Randpunkt u_0 nächsten Randpunkte jeweils rechts und links auf der zu u_0 benachbarten, horizontalen Gitterlinie, in Abb.9.2.2d überhalb von u_0. Die gleichen Überlegungen gelten wieder für die beiden Nachbardefekte von u_0, und spiegelsymmetrisch, auch für die Randpunkte bei u_1. Die Randpunkte ergeben damit als Form des r-maximalen Teilgraphen die Form einer Raute mit u_0 und u_1 als untere und obere Eckpunkte .



e) $r=r_k$ f) $r=r_k+2$

Abb.9.2.2e,f Beispiele des r-maximalem Teilgraphen

Für die Lage eines 'Mittelpunkts' der Raute sind zwei Fälle zu unterscheiden:

α) r ist gerade, r/2 ist gerade : $r=r_k$

Dann ist $r_1=r_2=id$ eine gerade Zahl und auf ID gibt es eine ungerade Zahl von Einheiten (S. Abb. 9.2.2e). Damit existiert eine Einheit M , die sowohl von u_0 als auch von u_1 den Abstand id/2 hat. Aus der Rotationssymmetrie um M folgt, daß M auch zu u_2 und u_3 den gleichen Abstand id/2 hat. M hat zu den horizontalen Gitterlinien der Nachbardefekte von u_0 und u_1 den Abstand id/2 -1. Alle Einheiten von id haben wiederum den Abstand 1 von der benachbarten vertikalen Gitterlinie, so daß M den Abstand id/2 auch zu den Nachbardefekten von u_0, u_1, und aus Symmetriegründen, auch von u_2 und u_3 hat. Da deren Nachbardefekte um die gleiche Kantenzahl, die ihre horizontale Gitterlinie an dem Schnittpunkt mit (u_0,u_1) 'näher' an M liegt, 'ferner' von M mit dem Schnittpunkt ihrer vertikalen Gitterlinie mit (u_2,u_3) sind, hat jeder Randpunkt den Abstand id/2 von der Einheit M.

Die Zahl der intakten Einheiten des Teilgraphen ergibt sich mit id:

v(id) = (Zahl der Einheiten von V_1 auf der Gitterlinie von (u_0,u_1))
+ 2(Zahl der Einheiten von V_1 auf den benachbarten Gitterlinien)

$$v(id) = id-1 +2(id-1-2) +2(id-1-4) +\ldots$$

$$v(id)= id-1 + 2 \sum_{i:=1}^{id/2\ -1} (id-1-2i)$$

$$= id-1 +2(id/2-1)(id-1) \ -4 \sum_{i:=1}^{id/2-1} i$$

$$= 1+ \frac{id^2}{2} -id \ = 1+ \frac{4}{8}id^2 - \frac{2}{2}id \ = 1+ \frac{r}{2}\left(\frac{r}{k(G)}-1\right)$$

ß) r ist gerade; r/2 ist ungerade: $r=r_k+2$

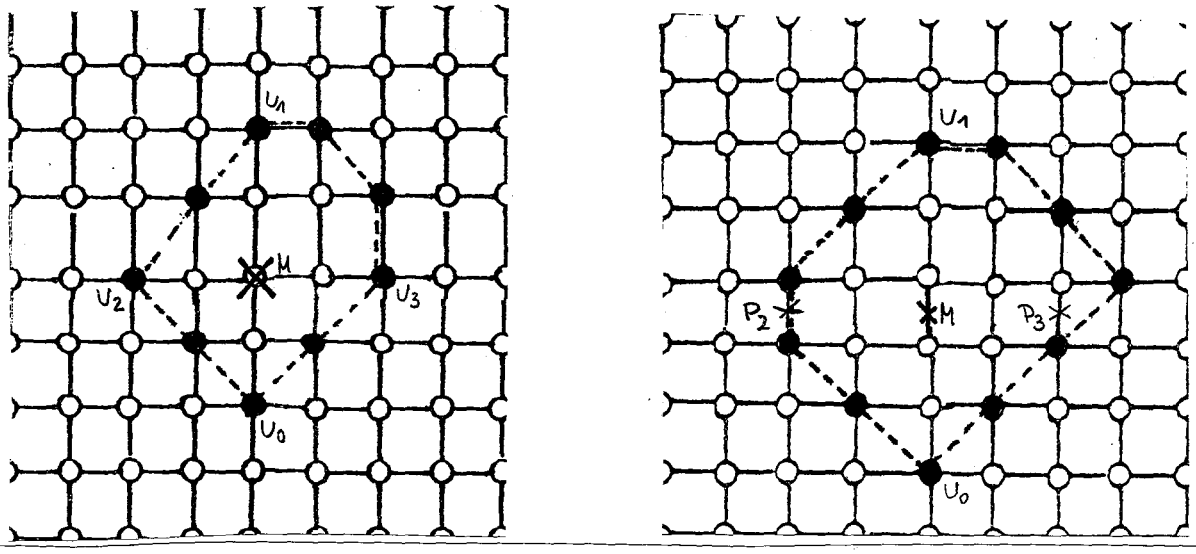
Dann ist $r_1=r_2=id$ eine ungerade Zahl und auf dem Weg ID gibt es eine gerade Zahl von intakten Einheiten (s. Abb. 9.2.2f). Angenommen, es ist möglich, zwischen der (id-1)/2 ten und (id-1)/2 +1 ten intakten Einheit auf halber Kantenlänge einen Punkt M zu fixieren. Dann hat dieser Punkt den Abstand id/2 zu u_0 und u_1 und ebenso zu u_2 und u_3. Mit den Argumenten aus α) hat dieser Punkt auch den gleichen, nicht ganzzahligen Abstand zu

allen Randpunkten. Für die Zahl der isolierten Einheiten betrachten wir in dem r-maximalen Teilgraph in Abb.9.2.2f die horizontale Gitterlinie neben der Geraden durch P_2, M, P_3. Deren Zahl von isolierten Einheiten ist die Differenz zu $v(r_0)$. Damit gilt

$v(r)= v(r_k+2)= v(r_k)+id-2 = v(r_k)+(r_k+2)/2 -2 = v(r_k)+r_k/2 -1.$

(2) Sei r ungerade; $r=r_k+1$ oder $r=r_k+3$.

Wie sieht der r-maximale Teilgraph aus, wenn zusätzlich zu den Defekten in Abb.9.2.2e,f noch eine weitere Einheit defekt ist?



g) $r=r_k+1$ h) $r=r_k+3$

Abb.9.2.2g,h r-maximale Teilgraphen bei ungeradem r

a) $r=r_k+1$

Die Menge der Defekten läßt sich wieder in zwei Teile unterteilen: einen Teil mit $r_0= \lfloor r/2 \rfloor$ und der andere Teil mit $r_1=\lceil r/2 \rceil$ Einheiten.

Dann ist das Teilstück des r-maximalen Teilgraphen in R_0 in Abb.9.2.2e identisch zu dem in Abb.9.2.2g.

Betrachten wir R_1. Da die Menge $R_1 \cup\{u_0\}$ eine gerade Zahl von Randpunkten enthält und die Zahl der horizontalen Gitterlinien von u_1 bis u_0 ungerade ist, müssen beim r-maximalen Teilgraphen zwei Defekte auf einer horizontalen Gitterlinie existieren. Falls der zusätzliche Defekt in Abb.9.2.2g auf einer horizontalen Gitterlinie überhalb u_1 oder unterhalb u_0 liegt, so werden keine Einheiten in G_1 zusätzlich isoliert. Da die Geometrie des Netzes spiegelsymmetrisch zur horizontalen Gitterlinie durch u_2,u_3 ist, reicht es nun, nur noch die horizontalen Gitterlinien von

u_1 bis u_3 zu betrachten.
Wie aus Abb.9.2.2g zu ersehen ist, werden genau dann am meisten Einheiten zusätzlich isoliert, wenn der zusätzliche Defekt auf der selben horizontalen Gitterlinie wie u_1 liegt und alle Defekte von R_1, die auf den horizontalen Gitterlinien zwischen u_1 und u_3 liegen, 'um einen Platz nach rechts rücken', d.h. auf der gleichen horizontalen, aber benachbarten, vertikalen Gitterlinie rechts daneben sich befinden. Dabei werden zusätzlich $r_k/4 -1$ Einheiten isoliert. Also ist

$$v(r_k+1)=v(r_k)+r_k/4 -1.$$

Der r-maximale Teilgraph in Abb.9.2.2g ist nicht eindeutig; durch die Netzsymmetrie gibt es 4 Varianten, die aber alle die gleiche Zahl von intakten Einheiten enthalten.

b) $r=r_k+3$
Die Überlegungen von $r=r_k+1$ gelten analog für den r-maximalen Teilgraph in Abb.9.2.2f; der eine zusätzliche Defekt befindet sich auf der selben horizontalen Gitterlinie wie u_1. Durch die damit mögliche 'Verschiebung' der Randpunkte aus R_1 auf den horizontalen Gitterlinien zwischen u_1 und P_3 werden zusätzlich $r_k/4 +1-1=r_k/4$ Intakte isoliert. Damit ist

$$v(r)=v(r_k+3)=v(r_k)+3r_k/4 -1.$$

Auch in diesem Fall sind wieder 4 verschiedene r-maximale Teilgraphen möglich, die aber alle die gleiche Zahl von Intakten enthalten.

B) SYMMETRISCHE TEILUNG

Bei der symmetrischen Teilung gehört u_0 auch zu R_1, so daß mit den Überlegungen von A)a) und b) id+1 Defekte in R_1 sind diese sich auf verschiedenen vertikal und horizontal versetzten Gitterlinien befinden.

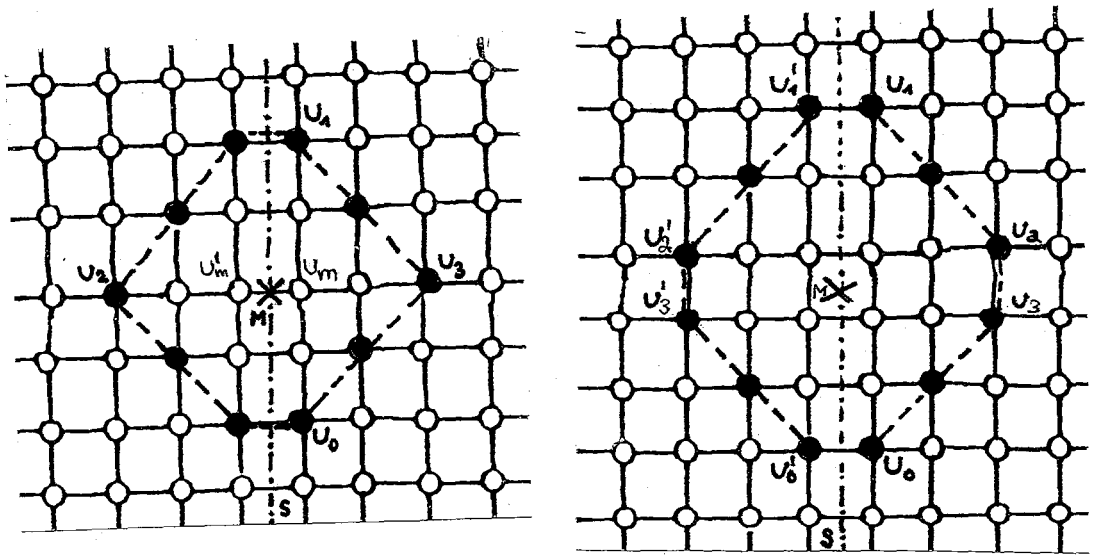


Abb.9.2.2i,j Beispiele von isolierten Teilgraphen

Für die Lage eines 'Mittelpunkts' der Teilgraphen sind zwei Fälle zu unterscheiden:

α) r/2 ist ungerade; also $r=r_k+2$.

Seien die entsprechenden Einheiten des linken Teils der symmetrischen Teilung mit einem ' notiert (s. Abb. 9.2.2i). Es ist $r_1=r_2=id+1$ eine gerade Zahl und auf ID gibt es eine ungerade Zahl von Einheiten. Dann existiert eine Einheit u_m, die sowohl von u_0 als auch von u_1 den Abstand id/2 hat. Da dies spiegelsymmetrisch in beiden Hälften des Teilgraphen der Fall ist, gibt es einen Punkt M, der von beiden Einheiten u_m und u_m' und damit auch von u_1, u_1' und u_0, u_0' den gleichen Abstand id/2 +1/2 hat. Aus der Rotationssymmetrie folgt, daß M auch zu u_2 und u_3 den gleichen Abstand hat. u_m hat zu den horizontalen Gitterlinien der Nachbardefekte von u_0 und u_1 den Abstand id/2 -1. Alle Einheiten von ID haben wiederum den Abstand 1 von der benachbarten vertikalen Gitterlinie, so daß u_m den Abstand id/2 auch zu den Nachbardefekten von u_0, u_1, und aus Symmetriegründen, auch von u_2 und u_3 hat. Da deren Nachbardefekte um die gleiche Kantenzahl, die ihre horizontale Gitterlinie an dem Schnittpunkt mit (u_0,u_1) 'näher' an u_m liegt, 'ferner' von u_m mit dem Schnittpunkt ihrer vertikalen Gitterlinie mit (u_2,u_3) sind, hat jeder Randpunkt den Abstand id/2 +1/2 von M.

Die Zahl der intakten Einheiten des Teilgraphen ergibt sich mit id:

$v(id)$ = 2(Zahl der Einheiten von V_1 auf der Gitterlinie von (u_0,u_1))
+ 2(Zahl der Einheiten von V_1 auf den benachbarten Gitterlinien)

$$v(id) = 2(id-1) +2(id-1-2) +2(id-1-4) +...$$

$$v(id)= 2 \sum_{i:=1}^{id/2} (id+1-2i)$$

$$= id^2+id -4 \sum_{i:=1}^{id/2} i = id^2+id-id^2/2-id = id^2/2$$

Mit $id=r/2 -1$ ist $v(r)= 1/2 +r/2(r/4 -1)$
Da $r/2$ ungerade ist, ist $r=r_k+2$ und somit ist
$v(r_k+2)=v(r)=v(r_k)+r_k/2 -1$

ß) $r/2$ ist gerade; also $r=r_k$.
Dann ist $r_1=r_2=id+1$ eine ungerade Zahl und auf ID gibt es eine gerade Zahl von intakten Einheiten (s. Abb. 9.2.2j). Angenommen, es ist möglich, zwischen der $(id-1)/2$ ten und $(id-1)/2$ +1 ten intakten Einheit auf halber Kantenlänge einen Punkt M zu fixieren. Dann hat dieser Punkt den Abstand $id/2$ zu u_0 und u_1 und ebenso zu u_2 und u_3. Mit den Argumenten aus α) hat dieser Punkt auch den gleichen, nicht ganzzahligen Abstand zu allen Randpunkten. Die Zahl der isolierten Einheiten ist

$$v_\beta(id)= 2(id-1) +2(id-1-2) +2(id-1-4) +...$$

$$v_\beta(id)= 2 \sum_{i:=1}^{(id-1)/2} (id+1-2i)$$

$$= id^2/2 - 1/2$$

Mit $id=r/2 -1$ und $r=r_k$ ist

$$v_\beta(r)= \frac{r^2}{8} - \frac{r}{2} -1 = v(r_k) -1$$

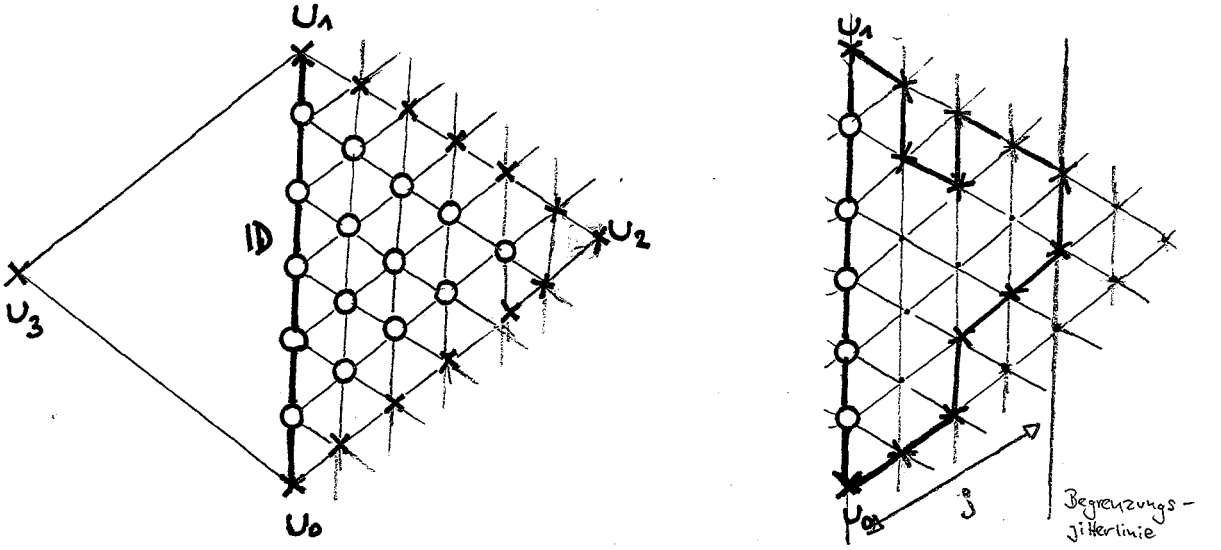
Dabei ist $v(r_k)= 1 + \frac{r_k^2}{8} - \frac{r_k}{2}$ die Zahl der in einem Teilgraphen nach Abschnitt A)(1) isolierten Einheiten. Da $v_\beta(r_k)<v(r_k)$ gilt, ist der symmetrisch teilbare Teilgraph mit r_k Defekten nicht r-maximal.

II) Das reguläre Flächennetz mit k(G) = 6

Sei r_k die Zahl, die durch 6 ohne Rest teilbar ist und die Gleichung $r=r_k+n$, $0\leq n<k=6$ erfüllt.
Die folgenden Untersuchungen sollen dazu dienen, die Form 6 verschiedener r-maximalen Teilgraphen zu bestimmen und die Zahl der Einheiten, die darin isoliert werden können, zu berechnen.

A) ASYMMETRISCHE TEILUNG

(1) Sei r/6 eine ganze Zahl: $r=r_k$.
Sei wieder ein Teilgraph G_1 durch r Defekte in einem 'ausreichend großen' G isoliert und sei G_1 r-maximal. Seien die Randpunkte wieder so aufgeteilt, daß $r_0=r_1=r/2$ Defekte in jedem Teil links und rechts von der Gitterlinie mit u_1 und u_0 sind. Da die Netzgeometrie spiegelsymmetrisch zu ID ist, reicht es wieder, nur eine Seite zu betrachten.



k) l)

Abb.9.2.2k,l Teilung des r-maximalen Teilgraphen

Um die Einheiten auf dem Weg ID nach dieser Seite in alle Richtungen zu isolieren, sind mindestens id+1 Defekte nötig, also $r_1\geq id+1$. Wieder wie in ii) ist ausgeschlossen, daß sich die Defekte auf einer Gitterlinie befinden, die in Abb.9.2.2k unterhalb von u_0 oder überhalb von u_1 die vertikale Gitterlinie von (u_0,u_1) schneidet, da sonst der Teilgraph nicht r-maximal wäre. Also kann der Nachbardefekt von u_0 nur auf einer Gitterlinie liegen, die ID oberhalb von u_0 schneidet und der von u_1 auf einer, die ID unterhalb von u_1 schneidet. Das gleiche Argument gilt aber

für den Nachbardefekt selbst und seinen Nachbardefekt. Also muß der r-maximale Teilgraph im Viereck liegen, das von u_0, u_2, u_1 und u_3 (s.Abb.9.2.2k) gebildet wird. Da nach Abb.9.2.2l jede Abweichung der Randpunkte von den dieses Viereck begrenzenden Gitterlinien auf eine benachbarte Gitterlinie bei gleicher Zahl von Defekten weniger intakte Einheiten isoliert, ist durch die Spiegelsymmetrie der Netzgeometrie um ID das Viereck oder, durch vertikale Gitterlinien zusätzlich begrenzt, das Sechseck die Form der Randpunkte des r-maximalen Teilgraphen.
Sei j der Abstand von ID zur vertikalen Gitterlinie, die den r-maximalen Teilgraph 'begrenzt' ('Begrenzungsgitterlinie'), d.h. Randpunkte von G_1 enthält und eine benachbarte Gitterline ohne Randpunkte besitzt (s.Abb.9.2.2h). Der Wert von j gibt die Form der Begrenzung an; vom schmalen Sechseck bei j=1 bis zum Viereck mit j=id. Wie groß ist j bei dem r-maximalen Teilgraphen?
Die Zahl v(id,j) der isolierten Einheiten ist

$$
\begin{aligned}
v(id,j) &= (\text{intakte Einheiten auf } (u_0,u_1)) \\
&\quad +2(\text{intakte Einheiten von } V_1 \text{ auf den benachbarten Gitterlinien}) \\
&= id-1 +2(id-2) +\ldots+2(id-1-(j-1)) \\
&= id-1 +2\sum_{i:=1}^{j-1}(id-1-i) \\
&= id-1\ 2id(j-1) -2(j-1) -2(j(j-1)/2) \\
&= 1 + 2idj -id -j^2-j
\end{aligned}
$$

Die Zahl der Randpunkte ist dabei

$$
\begin{aligned}
r(j) &= u_0+ u_1+ 4*\text{Nachbardefekte} +\ldots \\
&\quad +2*(\text{Randpunkte der vertikalen Begrenzungsgitterlinien}) \\
&= 2 +4j +2id -2 -2j = 2(id+j) \quad \text{oder } id=r/2 -j \qquad (*)
\end{aligned}
$$

Damit ist

$$
\begin{aligned}
v(r,j) &= 1+2(r/2-j)j -(r/2-j) -j^2 -j \\
&= 1 +rj -r/2 -3j^2
\end{aligned}
$$

oder

$$
\begin{aligned}
v(r,id) &= 1+ r(r/2-id) -r/2 -3(r/2-id)^2 \\
&= 1 -3id^2 +2rid -r^2/4 -r/2 \qquad (**)
\end{aligned}
$$

Das Maximum ist erreicht bei j* und id*:

$$\left.\frac{\partial v(r,j)}{\partial j}\right|_{j=j^*} = 0 \text{ oder } r=6j^*.$$

Also ist id*=3j*-j*=2j* oder j*=id*/2 und id*= r/3. (***)
Die Zahl der isolierten intakten Einheiten ist

$$v(r,j^*)=1+rj\ -r/2\ -3j^{*2}=\ 1\ +\ r^2/12\ -r/2$$

Mit k(G)=6 gilt damit

$$v(r)=\ 1+r/2(r/k(G)\ -1),\ \text{Q.E.D.}$$

Der r-maximale Teilgraph hat somit eine 'Breite', d.h. Abstand der beiden vertikalen Begrenzungsgitterlinien, von $2j^*=id$. Es gibt mit den zu k(G)=4 analogen Überlegungen eine Einheit, die zu allen Randpunkten den Abstand id/2 hat; die Randpunkte liegen alle auf einem sechseckigen 'Kreis' des Radius id/2. In Abb.9.2.2m ist ein solcher r-maximaler Teilgraph für r=12 zu sehen.

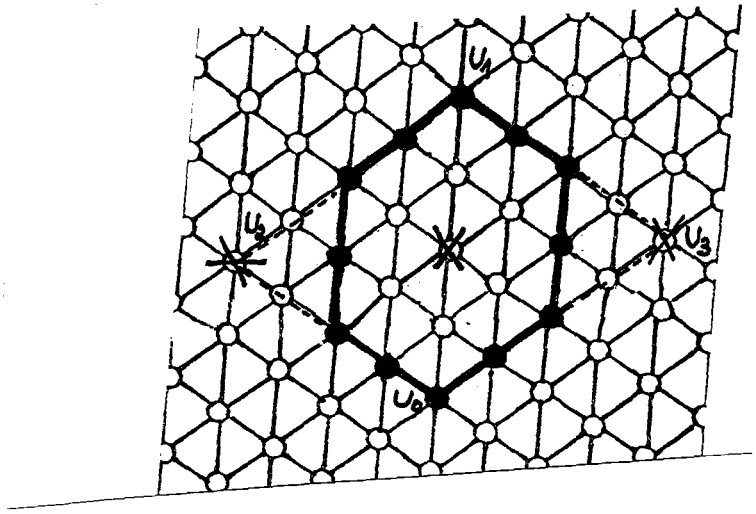


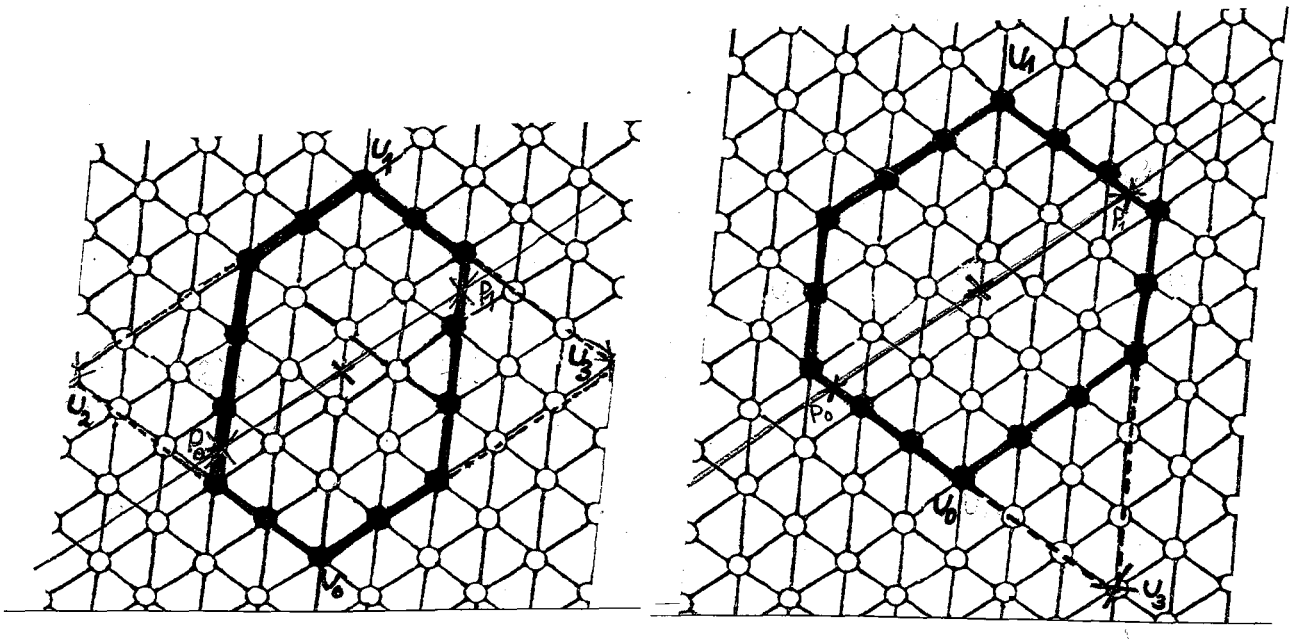
Abb.9.2.2m r-maximaler Teilgraph bei $r=r_k=12$

(2) r/6 ist keine ganze Zahl; r/2 ist eine ganze Zahl:
$r=r_k+2$ oder $r=r_k+4$
Da r eine gerade Zahl ist, läßt sich die Menge der Defekte wieder in zwei Teilmengen R_0 und R_1 aufspalten. Da die Netzstruktur spiegelsymmetrisch um ID ist, reicht es wieder, nur eine Seite des r-maximalen Teilgraphen zu betrachten. Mit den Argumenten aus (1) ist klar, daß die Randpunkte des r-maximalen Teilgraphen die Form eines Sechsecks bilden. Mit gegebenen id ist nach Formel (*) der Abstand der vertikalen Begrenzungsgitterlinie von ID mit j=r/2-id gegeben. Wie groß ist id beim r-maximalen Teilgraphen? Differenzieren wir dir Funktion v(r,id), die auch für (2) gilt, nach id und setzen die Ableitung gleich null, so erhalten wir die gleiche Optimalitätsbedingung (***). Beim r-maximalen Teilgraph ist also id*=r/3.

Wenn r/3 keine ganze Zahl ist, so kann diese Bedingung sicher nicht erfüllt sein. Stattdessen gibt es zwei Teilgraphen, deren id die Maximumsbedingung annähernd erfüllen: einen mit $id_1:=\lceil r/3 \rceil$ und einen mit $id_2:=\lfloor r/3 \rfloor$. Welcher isoliert die meisten Einheiten?
Es ist mit (**)

$$v(r_k+2,id_1)-v(r_k+2,id_2)=$$
$$3(id_2^2-id_1^2)+2r(id_1-id_2)= r_k+1 > 0$$

a) $r=r_k+2$
Der r-maximale Teilgraph hat einen Internode-Durchmesser $id=\lceil r/3 \rceil = r_{k/3}+1$, einen Abstand des ID von der vertikalen Begrenzungsgitterlinie von $j=r_k/2 - r_k/3$ und isoliert $v(r,id)=v(r_k)+r_k/3-1$ intakte Einheiten (Abb.9.2.2h).



n) $r=r_k+2$ o) $r=r_k+4$
Abb 9.2.2n,o r-maximale Teilgraphen bei $r_k=12$

b) $r=r_k+4$
Es ist $id_1:=\lceil r/3 \rceil = r_k/3 +2$, $id_2:=\lfloor r/3 \rfloor = r_k/3 +1$. Damit ist $v(r,id_1)-v(r,id_2)=-1<0$, so daß der r-maximale Teilgraph mit $r=r_k+4$ Defekten einen id von $r_k/3 +1$, einen Abstand $j=r_k/2 - r_k/3 +1$ des ID von der vertikalen Begrenzungsgitterlinie besitzt und $v(r,id)=v(r_k)+ 2r_k/3 -1$ intakte Einheiten isoliert (Abb.9.2.2o).

(3) r/6 ist keine ganze Zahl, r/2 ist keine ganze Zahl:

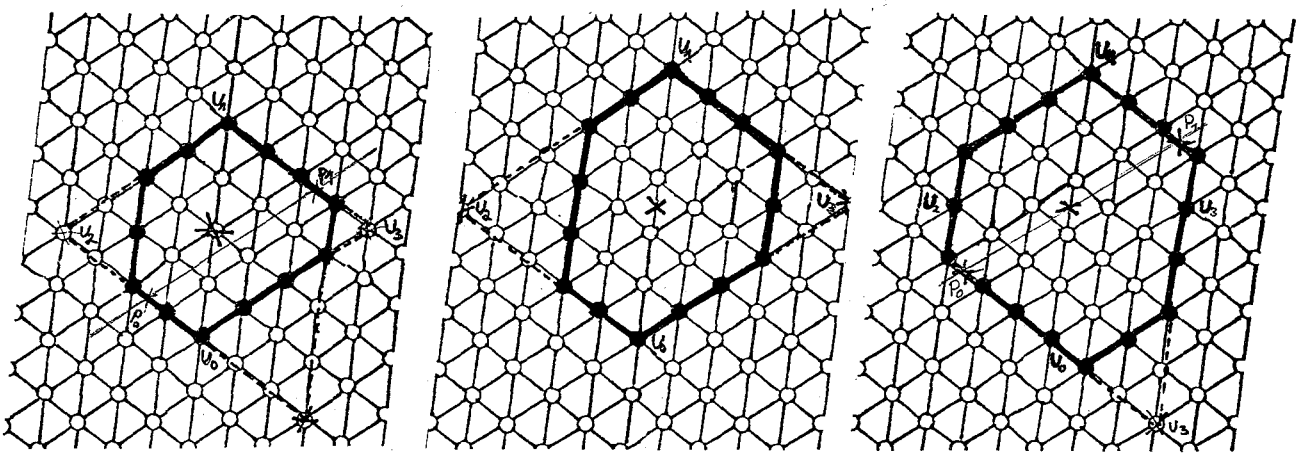
$r=r_k+1$, $r=r_k+3$, $r=r_k+5$

Betrachten wir zunächst Abb.9.2.2l. Die Lage der vertikalen Begrenzungsgitterlinie im Dreieck u_1, u_3, u_0 wird von der Zahl der Randpunkte bestimmt. Bei einem Randpunkt zusätzlich ist ein r-maximaler Teilgraph G_1' möglich, dessen vertikale Begrenzungsgitterlinie in Abb.9.2.2l die benachbarte Begrenzungsgitterlinie von G_1 darstellt. Seien auf der Begrenzungsgitterlinie von G_1 r' Randpunkte, so können mit einem zusätzlichen Randpunkt r'-2 intakte Einheiten zusätzlich in G_1' isoliert werden.

Betrachten wir nun den r-maximalen Teilgraphen mit $r=r_k$ in Abb.9.2.2m. Da alle 6 Begrenzungsgitterlinien gleich viele Randpunkte besitzen, gibt es 6 Möglichkeiten, mit einem zusätzlichen Randpunkt $r'-2=(r_k/6+1)-2=r_k/6-1$ Einheiten zusätzlich zu isolieren. Also ist

$$v(r)=v(r_k+1)=v(r_k)+r_k/6-1.$$

In Abb.9.2.2p ist ein solcher r-maximaler Teilgraph zu sehen.



p) $r=r_k+1$ q) $r=r_k+3$ r) $r=r_k+5$

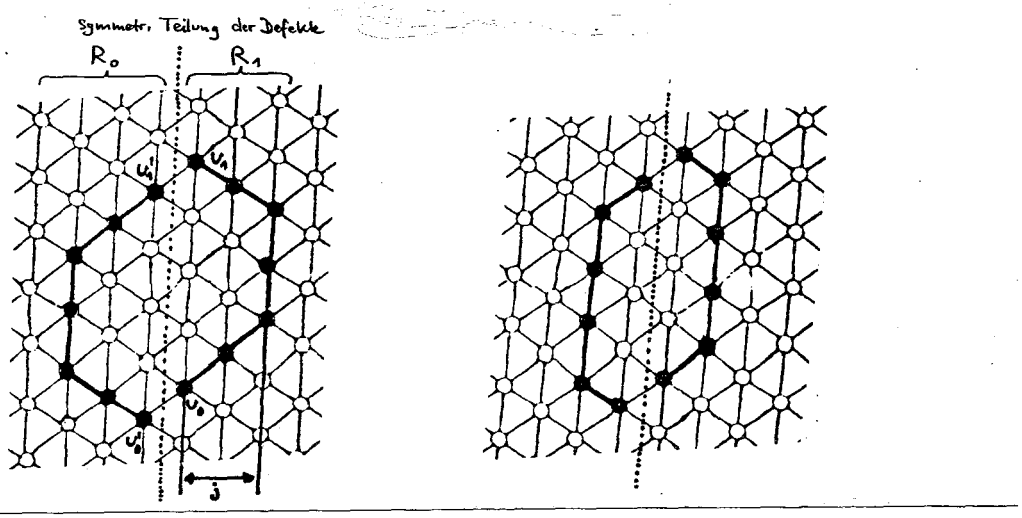
Abb.9.2.2p,q,r r-maximale Teilgraphen bei $r_k=12$

Werden obige Überlegungen auf die r-maximalen Teilgraphen mit $r=r_k+2$, $r=r_k+4$ angewendet, so isolieren die r-maximalen Teilgraphen mit einem zusätzlichen Randpunkt zusätzlich $r_k/6$ intakte Einheiten, da die maximale Zahl von Randpunkten auf einer Begrenzungsgitterlinie jeweils $r_k/6+2$ ist (Abb.9.2.2q,r).

Damit ist die Gleichung $v(r_k+n)=v(r_k)+nr_k/6-1$ für alle n=1,2,3,4,5 bewiesen.

B) SYMMETRISCHE TEILUNG

Seien die beiden auf g_1 zur vertikalen Isolation nötigen Defekte mit u_0 und u_1, die auf g_0 mit u_0' und u_1' bezeichnet. Da G_1 isoliert ist, müssen u_0 und u_0', ebenso wie u_1 und u_1', zusammenhängen. Sei o.B.d.A. u_1 in Abb.9.2.2s 'höher' als u_1' gelagert.



s) t)

Abb 9.2.2s,t Beispiele isolierter Teilgraphen bei symmetrischer Teilung

Je nach Internode-Durchmesser id und der Zahl r der Defekte enthält der rechte Teil von G_1 eine bestimmte, maximale Zahl von intakten Einheiten. id_1 ist so gegeben, daß diese Zahl maximal bei gegebenem r_1 ist (s. Teil A)(1)). Wäre nun u_0' höher in der Abbildung als u_0, so wäre id_2 des linken Teils kleiner als id_1. Damit würden weniger Einheiten als möglich in G_1 isoliert, im Widerspruch zur Vorraussetzung. Also ist $id_1=id_2$; der linke Teil der Randpunkte bildet spiegelsymmetrisch die gleiche Form wie der rechte Teil, nur 'versetzt' nach links unten.
Die Zahl der isolierten Einheiten ist mit j, dem Abstand von ID_1 und ID_2 zur jeweiligen Begrenzungsgitterlinie

$$v(id,j) = 2(id-1) +2(id-2)+\ldots+2(id-1-(j-1))$$

$$= 2 \sum_{i=1}^{j} (id-1-(i-1)) \quad = 2 \sum_{i=1}^{j} (id-i) = -j^2 +j(2id-1)$$

Da in der gleichen Konfiguration in A)(1) in (*) zwei Defekte weniger berücksichtigt wurden als hier, ist $(r-2)/2 -j=id$ oder $j=r/2 -id-1$.
Somit ist

$$v(r,id) = -\frac{r^2}{4} + 2idr + \frac{r}{2} - 3id^2 - 3id$$

Das Maximum ist erreicht, wenn die Ableitung von v(r,id) nach id null ist bei id=id*. Dies ist der Fall bei

$$id^* = r/3 - 1/2$$

Da id* im Unterschied zu der rechten Seite der Gleichung nur ganzzahlig sein kann, sind zwei verschiedene, submaximale id* möglich:

$$id_1^* := \lfloor r/3 - 1/2 \rfloor$$

und

$$id_2^* := \lceil r/3 - 1/2 \rceil .$$

Im Folgenden wird für $r=r_k$, $r=r_k+2$ und $r=r_k+4$ untersucht, bei welchem id* die meisten Einheiten isoliert werden.

a) $r=r_k$

$r_k/3$ ist nach Definition eine ganze Zahl. Also ist

$$id_2^* = \frac{r_k}{3}, \quad v(r,id_1^*) = \frac{r_k}{2}\left(\frac{r_k}{6} - 1\right) = v(r_k) - 1$$

mit $v(r_k)$ aus Teil A)(1).

$$id_1^* = \frac{r_k}{3} - 1, \quad v(r,id_2^*) = \frac{r_k}{2}\left(\frac{r_k}{6} - 1\right) - \frac{4}{3}r_k = v(r_k) - \frac{4}{3}r_k - 1$$

Ein symmetrischer Teilgraph isoliert bei $r=r_k$ Defekten immer weniger Einheiten als der maximale, asymmetrische Teilgraph und ist deshalb nicht r-maximal bei $r=r_k$. Ein Beispiel mit $r_k=12$ zeigt Abbildung 9.2.2t.

b) $r=r_k+2$

$$id_1^* = \frac{r_k}{3}, \quad v(r,id_1^*) = \frac{r_k}{2}\left(\frac{r_k}{6} - 1\right) + \frac{1}{3}r_k = v(r_k) + \frac{1}{3}r_k - 1 = \lfloor v(r) \rfloor$$

$$id_2^* = \frac{r_k}{3} + 1, \quad v(r,id_2^*) = \frac{r_k}{2}\left(\frac{r_k}{6} - 1\right) + \frac{1}{3}r_k - 2 = v(r_k) + \frac{1}{3}r_k - 3$$

G_1 enthält mit id_1^* die maximale Zahl von Einheiten, die identisch ist mit der Behauptung von Lemma 9.2b.

Der maximale Teilgraph bei symmetrischer Teilung ist identisch mit dem maximalen Teilgraph bei asymmetrischer Teilung in Abbildung 9.2.2.n und damit r-maximal; die symmetrische Teilungslinie ist die Gerade durch die Punkte P_0 und P_1.

c) $r=r_k+4$

$$id_1^* = \frac{r_k}{3}, \quad v(r,id_1^*) = \frac{r_k}{2}\left(\frac{r_k}{6} - 1\right) + \frac{2}{3}r_k - 2 = v(r_k) + \frac{2}{3}r_k - 3$$

$$id_2^* = \frac{r_k}{3}+1,\ v(r,id_2^*) = \frac{r_k}{2}\left(\frac{r_k}{6}-1\right)+\frac{2}{3}r_k = v(r_k)+\frac{2}{3}r_k-1 = \lfloor v(r)\rfloor$$

G_1 enthält mit id_2^* die maximale Zahl von Einheiten, die identisch ist mit der Behauptung von Lemma 9.2b.

Der maximale Teilgraph bei symmetrischer Teilung ist wieder identisch mit dem maximalen Teilgraph bei asymmetrischer Teilung in Abbildung 9.2.2.o und damit r-maximal; die symmetrische Teilungslinie ist die Gerade durch die Punkte P_1 und P_0.

Also ist auch bei symmetrischer Teilung Lemma 9.2b erfüllt, Q.E.D.

III) Isolation

Im bisher vorgestellten Beweis mit den Teilen I) und II) wurde bewiesen, daß bei r Defekten maximal $v(r)=1+(r/2)(r/k(G)-1)$ intakte Einheiten in einem Teilgraph kommunikationsmäßig von den anderen Einheiten in G abgeschnitten sind. Zur Eigenschaft der Isolation gehört aber nach der Definition auch noch die Bedingung $v_1 \leqq v-v_0-v_1$ ('Die isolierte Menge ist kleiner als die Restmenge').

Sei k(G) mit k notiert.

Aus $t>1$ und $(k-4)/4k \geqq 0$ folgt

$$t^2(1/4-1/k)+t-1 > 0$$
$$(t^2/4+t+1)-t > 2-t+t^2/k$$
$$(t/2+1)^2-t > 2(1+t/2(t/k-1))$$

Mit $N>(t/2+1)^2$ ist

$$N-t > 2\,v(t)$$

$N-v(t)-t=v' > v(t)$, die Bedingung der Isolation erfüllt.

9.2.3 BEWEIS von Lemma 9.2c:

a) Sei kein Teilgraph isoliert.
Bei $r<k(G)$ können keine intakten Einheiten isoliert werden; der r-maximale Teilgraph enthält null Einheiten, Q.E.D.
Seien r Einheiten mit $r\geq k(G)$ im System vorhanden. Da k(G) hier die Zahl der Nachbarn ist, könnten die r Defekte auch so verteilt sein, daß mindestens eine intakte Einheit isoliert ist, der r-maximale Teilgraph also mehr als null Einheiten hat, Q.E.D.

b) Sei mehr als ein Teilgraph isoliert.
Betrachten wir zwei der isolierten Teilgraphen; beispielsweise G_i und G_j. Nehmen wir an, die Teilgraphen sind r_1-maximal, d.h. jeder durch r_1 Defekte isolierte Teilgraph G_1 hat nach Lemma 9.2b $v(r_1)= 1+(r_1/2)(r_1/k(G)-1)$ Einheiten; mehr als jeder andere, nicht r_1-maximale, isolierte Teilgraph.
Dann lassen sich zwei Fälle unterscheiden, je nachdem, ob die Defekte Randpunkte nur eines oder mehrerer Teilgraphen gleichzeitig sind.

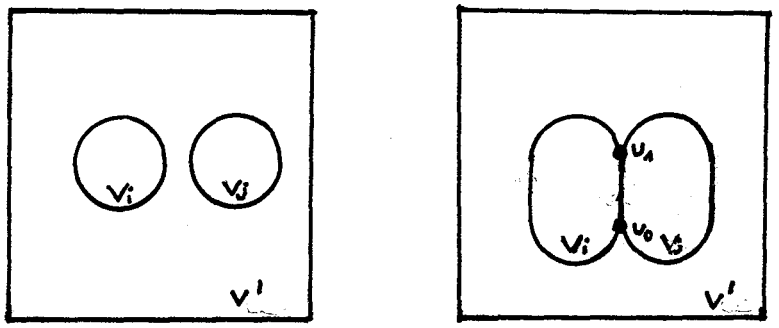


Abb. 9.2.3 a) b)

Sei in den folgenden Rechnungen k(G) mit k abgekürzt.

i) Es gibt keinen Defekt, der gleichzeitig (s.Abb.9.2.3a) Randpunkt der zwei isolierten Teilgrapen G_i und G_j ist. Sei $r_m:=r_i+r_j$.
Betrachten wir den ungünstigen Fall, wenn r_i und r_j durch k ohne Rest teilbar, dagegen r_m nicht. Dann sind durch r_m Defekte immer mehr Intakte in einem Teilgraphen G_m isoliert als in G_i und G_j zusammen, da mit $r_m=r_i+r_j$ folgt

$$v_i(r_i) + v_j(r_j) = 2 + \frac{r_i^2}{2k} + \frac{r_j^2}{2k} - \frac{r_i}{2} - \frac{r_j}{2}$$

$$v_m(r_i+r_j) = \left[1 + \frac{r_i^2}{2k} + \frac{r_i r_j}{k} + \frac{r_j^2}{2k} - \frac{r_i}{2} - \frac{r_j}{2} \right] = v_i(r_i)+v_j(r_j) + \frac{r_i r_j}{2} - 2$$

Da $k>2$ ist $r_i r_j \geq k^2 > 2k$ und somit $v_m > v_i(r_i)+v_j(r_j)$.

ii) Es gibt Defekte, die sowohl Randpunkt von G_i als auch von G_j sind. Seien die beiden Teilgraphen so angeordnet, daß sie möglichst viele Defekte als gemeinsame Nachbarn haben, z.B. mit einer gemeinsamen Seite (u_0,u_1) in Abb.9.2.3b.
Angenommen, auf dem Weg zwischen u_0 und u_1 liegt mindestens ein Defekt. Dann sind in dem einen Teilgraphen G_m, der dadurch entsteht, daß alle Defekte zwischen u_0 und u_1 nicht defekt sind, bei geringerer Zahl von Defekten in G_m mehr Einheiten isoliert als in beiden Teilgraphen G_i und G_j zusammen.
Gibt es dagegen keine Defekte zwischen u_0 und u_1 und r_i ist r_i-maximal, r_j ist r_j-maximal, so sind nur die beiden Konfigurationen in Abb.9.2.3c,d möglich. Wie aus der Abbildung zu ersehen ist, ist v_m auch hier gleich oder größer als v_i+v_j.

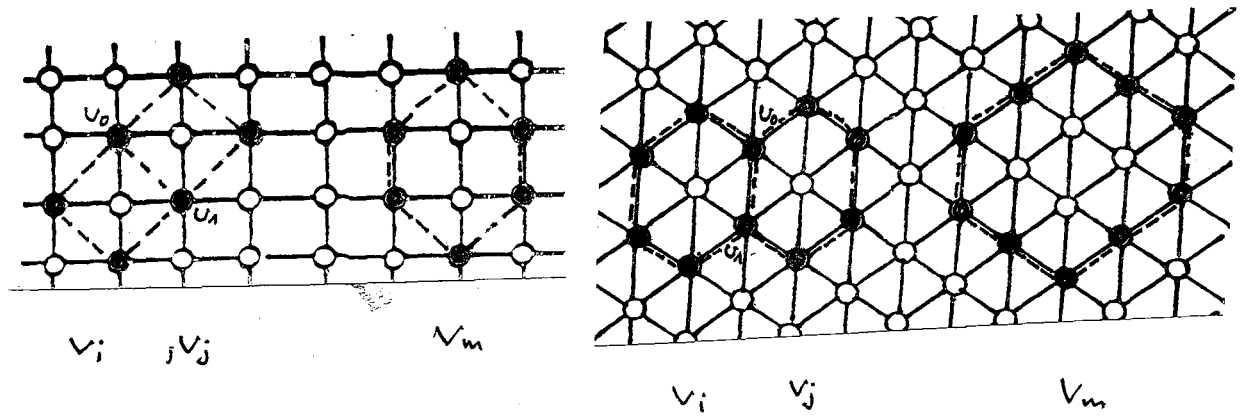


Abb. 9.2.3 c)k=4 d)k=6

Die Argumente i) und ii) gelten für beliebige G_i und G_j. Seien nun alle isolierten Teilgraphen numeriert $G_1 \ldots G_n$ mit n 1. Dann existiert ein Teilgraph G_1^1, in dem bei allen Defekten, die G_1 und G_2 isolieren,

mehr Intakte isoliert werden als in G_1 und G_2 zusammen. Ebenso existiert ein Teilgraph G_1^2, in dem bei allen Defekten, die G_1^1 und G_3 isolieren, mehr Intakte isoliert werden. Dies läßt sich schrittweise bis G_n durchführen, so daß es schließlich einen isolierten Teilgraphen G_1^{n-1} gibt, der bei gleicher Zahl von Defekten von G_1^{n-2} und G_n mehr Intakte isoliert als G_1^{n-2} und G_n zusammen, Q.E.D.

10. Ausblick

In Kapitel 4 wurde als allgemeines Diagnoseverfahren die Bayesdiagnose gewählt und in Kapitel 6 dezentrale Diagnosealgorithmen dazu formuliert. Dabei blieben aber noch viele Fragen ungeklärt. Beispielsweise gibt es trotz einiger Ansätze /FUJ1/ noch kein Verfahren, um bei N Einheiten mit unterschiedlichen Zuverlässigkeiten R_i einen Testgraphen so zu konstruieren, daß er bei unvollständigen Tests maximale Diagnostizierbarkeit besitzt.
Auch sind die für eine Wahrscheinlichkeits- Diagnose nötigen Parameter wie die Zuverlässigkeit R_i und die Fehlererkennungs- Wahrscheinlichkeit der Tests schwierig zu bestimmen, da sie sich u.U. während der 'Lebensdauer' der Diagnoseprogramme ändern können. Die Ansätze, diese Parameter bei der Diagnose zu lernen /BRA1/, müssen noch weiter voran getrieben werden; ebenso die Bestrebungen, den hohen Komplexitätsgrad der Diagnosealgorithmen der Bayesdiagnose zu erniedrigen, da die Diagnose sonst für die Computeranwendung unpraktikabel ist. Ein Ansatz in diese Richtung stellt die lokale Diagnose aus Kapitel 8 dar, doch auch hier fehlen theoretische Untersuchungen, unter welchen Umständen eine Erweiterung des Testgraphen sinnvoll ist.

Ein noch teilweise ungeklärtes Problem ist auch die Diagnose in VLSI-Multiprozessorsystemen. Kapitel 6 und 8 setzen für jeden Prozessor eines solchen Netzes Fähigkeiten voraus, die denen eines kompletten Mikrocomputers entsprechen, obwohl die 'systolischen Felder' in der VLSI-Konzeption ihrer Autoren nur Rechenelemente mit sehr beschränkten Möglichkeiten enthalten. Die für solche Systeme geeigneten Diagnosealgorithmen sind aber bisher kaum untersucht worden.

Anhang A

Die Komplexität der probabilistischen Diagnose und der Bayesdiagnose

Die probabilistische Diagnose sucht die Fehlerklasse $\sigma(S)$ bei N Einheiten und M Tests im System mit

$$P(S,\sigma(S)) = \max_{\substack{i \\ i=0,\ldots,2^N-1}} P(S,F_i)$$

und

$$P(S,F_i) = P(S/F_i)P(F_i) = \prod_{j:=0}^{M-1} P(t_j/F_i) \prod_{k:=0}^{N-1} R_k^{1-e_{ik}} (1-R_k)^{e_{ik}}$$

$$e_{ik} := \begin{cases} 0 & u_k \text{ defekt in } F_i \\ 1 & u_k \text{ intakt in } F_i \end{cases}$$

Die Bayesdiagnose sucht diejenige Fehlerklasse, die das geringste Risiko

$$r_{Bay}(S)) = \min_{\substack{k \\ k=0,\ldots,2^N-1}} r_k(S)$$

mit

$$r_k(S) = \sum_{i=0}^{2^N-1} L_{ik} P(S/F_i)P(F_i)$$

hat.

Sei die Komplexität eines Algorithmus durch die Größenordnung der Zahl der Operationen gegeben, bei gegebenen Ausgangsdaten ihn auszuführen. Sei analog dazu die Speicherkomplexität eines Algorithmus als die Größenordnung der Zahl der zur Ausführung nötigen Speicherplätze bezeichnet.

Viele arithmetischen Ausdrücke lassen sich entweder für jeden benötigten Parameter vor der Diagnose ausrechnen und als Listen führen (konstante Ausführungszeit) oder aber bei jeder Benutzung mit den aktuellen Parametern neu

errechnen (konstanter Speicherbedarf). Selbstverständlich läßt sich auch beides miteinander kombinieren; zur Charakterisierung der Diagnosen soll dies aber nicht betrachtet werden.
Betrachten wir nun die Komplexität der beiden Diagnosen bezüglich der N Einheiten und der M Tests im System.

Seien die Fehlerklassen und die Syndrome jeweils die dezimalen Äquivalente der binären Tupel $(u_{N-1},\ldots,u_0)$ und $(t_{M-1},\ldots,t_0)$, so daß keine besondere Berechnungszeit oder spezieller Speicherplatz dafür kalkuliert werden muß. Seien die Speicherplätze für den Testgraphen und die N Zuverlässigkeitskoeffizienten R_i nicht mitbetrachtet. Dann wird $P(F_i)$ als Produkt von N Faktoren aus den N Koeffizienten R_i errechnet oder als Liste von 2^N Wahrscheinlichkeiten geführt. Bei $P(S/F_i)$ ist ein Produkt aus M Faktoren zu errechnen oder eine Liste zu führen, die bei jeder der 2^N Fehlerklassen für jedes der 2^M Syndrome einen Wahrscheinlichkeitswert enthält.
Die probabilistische Diagnose sucht das Maximum von 2^N Verbundwahrscheinlichkeiten $P(S,F_i)$, wobei für jedes $P(S,F_i)$ maximal N+M-1 $\approx$ N+M Multiplikationsoperationen nötig sind. Ist einer der Faktoren Null, so kann die Multiplikation auch vorher mit dem Ergebnis Null abgebrochen werden. Ist die prob. Diagnose als Listenentscheidung implementiert, so ist bei konstanten Parametern eine Diagnoseentscheidung für jedes der 2^M möglichen Syndrome gespeichert.
In die Bayesdiagnose gehen noch zusätzliche Kostenkoeffizienten L_{ik} ein. Diese Koeffizienten lassen sich entweder aus der Angabe der Fehlerklassen durch Vergleich des diagnostizierten und des tatsächlichen Zustands in N Schritten errechnen, oder aber für jede der 2^N Fehlerklassen als Liste von 2^N Kostenkoeffizienten speichern.
Das Risiko einer Diagnose wird als Erwartungswert mit 2^N Additionen von einem Produkt der Zeitkomplexität von jeweils N+N+M Operationen errechnet. Alternativ dazu kann das Risiko auch als Liste für jede der 2^M Syndrome und jede der 2^N Fehlerklassen eine Kostenzahl enthält.
Die Bayesdiagnose sucht nun aus den 2^N Risiken das Minimum, wobei jedes Risiko selbst wieder mit der Zeitkomplexität $O(2^N(2N+M))$ errechnet wird. Damit hat die Bayesdiagnose die hohe Komplexität $O(2^{2N}(2N+M))$, oder alternativ, eine Liste, in der für jedes der 2^M Syndrome einer Fehlerklasse eingetragen ist. In der nachstehenden Tabelle ist dies zusammengefaßt:

Ausdruck	Zeitkomplexität bei konst.Speicher	Speicherkomplexität bei konst. Ausführungszeit
$P(F_i)$	$O(N)$	$O(2^N)$
$P(S/F_i)$	$O(M)$	$O(2^{M+N})$
prob.Diagnose	$O(2^N(N+M))$	$O(2^M)$
Kosten L_{ik}	$O(N)$	$O(2^{2N})$
$r_k(S)$	$O(2^N(2N+M))$	$O(2^{N+M})$
Bayesdiagnose	$O(2^{2N}(2N+M))$	$O(2^M)$

Ein Beispiel soll die hohe Komplexität der Bayesdiagnose veranschaulichen. In Kapitel 8 wird das erwartete Risiko einer Bayesdiagnose als Summe der erwarteten Risiken pro Syndrom über alle 2M Syndrome errechnet. Die Komplexität dieses Ausdrucks ist also $O(2^M 2^{2N}(2N+M))$ oder $O((2N+M)2^{2N+M})$. Die Ausweitung des Testgraphen geschieht in n Stufen. Angenommen, als System sei ein lineares System aus Abb.8.1 betrachtet, so ist $M=4n-2$ und $N=2n+1$. Der obige Ausdruck hat also eine von der n-ten Stufe abhängige Komplexität von $8n2^{8n}$. Auf einem Microcomputer ('Microengine') benötigt die Rechnung bei n=1, also 2^{11} Operationen, eine Zeit von 2 sec. Bei n=2 werden schon 2^{20} Operationen, also 17 Minuten, benötigt und bei n=3 sind es $3*2^{27}$ Operationen oder 4,5 Tage. Hierbei zeigen sich deutlich der Nutzen einer Komplexitätsabschätzung und die Nachteile einer allgemeinen Bayesdiagnose: Es ist nicht sinnvoll, das mittlere Risiko einer Bayesdiagnose für linearen Testgraphen und n=4 zu errechnen, da man dafür maximal 2^{37} Operationen oder mehr als 5 Jahre warten müßte.

Referenzliste

/ADH/ M.Adham, A.D.Friedman
Digital System Fault Diagnosis
Journal of Design Automation and Fault-Tolerant Computing
V1 pp.115-132 Jan 1977

/ALL/ F.J.Allan, T.Kameda, S.Toida
An Approach to the Diagnosability Analysis of a System
IEEE Transactions on Computers Oct 1975

/AMM/ E.Ammann, M.Dal Cin
Efficient Algorithms for Comparison-Based Self-Diagnosis
Self-Diagnosis and Fault-Tolerance
Attempto Verlag Tübingen 1981

/AMM2/ E.Ammann
Modelle für die Selbstdiagnose fehlertoleranter Systeme
Diss. der Fakultät für Physik
Universität Tübingen 1983

/ARM/ J.Armstrong, G.Gray
Fault Diagnosis in a Boolean n-Cube Array of Microprocessors
IEEE Transactions on Computers Vol C30, Aug.1981

/ATT/ E.Ammann, R.Brause, M. Dal Cin, E.Dilger, J. Lutz, T. Risse
ATTEMPTO: A Fault-tolerant Multiprocessor Working Station;
Design and Concepts
FTCS-13, Milano 1983

/BAR1/ F.Barsi, F.Grandoni, P.Maestrini
A Theory of Diagnosibility of Digital Systems
IEEE Transactions on Computers Vol C25 June 1976

/BAR2/ F.Barsi
Probabilistic Syndrom Decoding in Self-Diagnosable Digital Systems
Digital Processes Vol 7, 1981
Georgi Publishing Company, CH-1813 St. Saphorin

/BEN1/ J.Bentley
A Parallel Algorithm for Constructing Minimum Spanning Trees
Journal of Algorithm 1, 1980, p.51-59

/BEN2/ J.Bentley, H.Kung
A Tree Machine foe Searching Problems
Proc. of the Int. Conf. on Parallel Processing
IEEE Aug 1979

/BLO1/ M.Blount
Probabilistic Treatment of Diagnosis in Digital Systems
Proc. of the FTCS-7 pp.72-77, June 1977

/BLO2/ _____
Modeling of Diagnosis in Fail-softly Computer Systems
Design Automation and Fault-tolerant Computing
Computer Science Press Inc., 1980

/BON/ G.Bonn, W.Heil, J.Kippe, F.Saenger
Selbsttest und Selbstrekonfiguration von Prozessrechnersystemen
am Beispiel des RDC-Systems
FhG -Berichte 1/2-79

/BOS/ D.Bossen, M.Hsiao
ED/FI: A Technique for Improving Computer System RAS
Proc. of the FTCS-11

/BRA1/ R.Brause, E.Dilger, T.Risse
Diagnosing Algorithm and Learning
Self-Diagnosis and Fault-Tolerance
Attempto Verlag Tübingen 1981

/BRA2/ R.Brause
Über die Realisierung fehlertoleranter Computer
Institutsmitteilungen, Inst. f.Informat., Tübingen 1981

/BUT/ J.T.Butler
Speed-Efficiency Complexity Trade-offs in
Universal Diagnosis Algorithm
IEEE Transactions on Computers C30, Aug.1981

/CIO/ P.Ciompi, L.Simoncini
Analysis and Optimal Design of Self-Diagnosable
Systems with Repair
IEEE Transactions on Computers C28/5 May 1979

/COY/ W. Coy
On the Design of Easily Testable Iterative Systems
of Combinatorial Cells
IEEE Transactions on Computers C28/5 May 1979

/DAL1/ M.Dal Cin, E.Dilger
Self-Diagnosis and Fault-Tolerance
Attempto-Verlag Tuebingen 1981

/DAL2/ M.Dal Cin
Fehlertolerante Systeme
Teubner-Verlag Stuttgart 1979

/DAL3/ M. Dal Cin, E.Dilger
On the Diagnosability of Self-Testing Multi-Microprocessor Systems
Microprocessing and Microprogramming 7, 1981

/DAV/ J. Davies
Clockarchitecture and management
Computer Architecture News 8/5 Aug 1981

/DE / B.De, H.Krakau
Fault-Tolerance in a Multiprocessor, Digital Switching System
IEEE Transactions on Reliability Vol R30/3, Aug 1981

/DIF1/ W.Diffie, M.Hellmann
New directions in Cryptography
IEEE Transactions on Inform. Theory IT 22/6, Nov 1976

/DIF2/ W.Diffie, M.Hellmann
Privacy and Authentication:
An Introduction to Cryptography
Proc. of IEEE Vol 67/3 Mar 1979

/DIL/ E.Dilger, T.Risse
Adaptive Selbst-Testende Systeme
GI- Fachtagung Fehlertolerierende Rechnersysteme
IFB 54 Springer-Verlag Berlin 1982

/DOL/ D. Dolev
The Byzantine General strikes again
Journal of Alg.3, 1982 Academic Press

/FÄR/ G.Färber
Taskspecific Implementation of Fault-Tolerance
in Process Automation Systems
Self-Diagnosis and Fault-Tolerance
Attempto Verlag Tübingen 1981

/FRI/ A.D.Friedman
A New Measure of Digital System Diagnosis
Proc. of the FTCS-5, pp.167-170 June 1975

/FUJ1/ H.Fujiwara, K.Kinoshita
Connection Assignments for Probabilistic Diagnosable Systems
IEEE Transactions on Computers C27, March 1978

/FUJ2/ _____
Some Existance Theorems for Probabilistically Diagnosable Systems
IEEE Transactions on Computers C27, Oct.1980

/GEY/ W. Geyer
32-Bit Mikrocomputer besitzt neuartige Architektur
Elektronik 5/1981

/GIL/ W.K.Giloi
Rechnerachitektur
Springer Verlag 1981

/HAK/ S.L.Hakimi, A.T.Amin
Characterization of Connection Assignment of Diagnosable Systems
IEEE Transactions on Computers, Jan 1974

/HAR/ F.Harary
Graph Theory
Addison-Wesley, 1969

/HAY/ J.Hayes
A Graph Model for Fault-Tolerant Computing Systems
IEEE Trans. on Computers Vol C 25/9 Sept 1976

/HOP/ A.Hopkins,B,Smith, J.Lala
FTMP- A Highly Reliable Fault-tolerant Multiprocessor
for Aircraft Control
Proc. IEEE Vol 66/10 Oct 1978

/INF/ System 9000
Inforex GmbH., Frankfurt

/JON/ A.Jones,P.Schwarz
Experience Using Multiprocessor Systems
Computer Surveys, Vol 12/2 June 1980

/KÄF/ H.Käfer
Fehlererkennung und Fehlerkorrektur bei HDDR-Magnetbandgeräten
Elektronik 5/1981

/KAM/ T.Kameda, S.Toida, F.Allan
A Diagnosing Algorithm for Networks
Information and Control 29, 1975
Academic Press

/KAR1/ S.Karunanithi, A.D.Friedman
Analysis of Digital Systems Using a New Measure of System Diagnosis
IEEE Transactions on Computers Vol C28 Feb.1979

/KAR2/ _____
System Diagnosis with t/s Diagnosability
Proc. of the Int. Symp. Fault-Tolerant Computing, June 1977

/KIM1/ C.Kime
An Analysis Model for Digital System Diagnosis
IEEE Transactions on Computers, Vol C19 Nov 1970

/KIM2/ J.Russell, C.Kime
System Fault Diagnosis: Closure and Diagnosibility with Repair
IEEE Transactions on Computers, Vol C24 Nov 1975

/KIM3/ J.Russell, C.Kime
System Fault Diagnosis: Masking,Exposure and Diagnosibility
without Repair
IEEE Transactions on Computers, Vol C24 Dec 1975

/KIM4/ J.McPherson, C.Kime
A two-level Diagnostic Model for Digital Systems
IEEE Transactions on Computers, Vol C27 Jan 1979

/KOR/ Israel Koren
A Reconfigurable and Fault-Tolerant VLSI Multiprocessor Array
Proc. of the FTCS-11, 1981

/KUH1/ J.Kuhl, S.Reddy
Distributed Fault-Tolerance for Large Multiprocessor Systems
Proc. of the 7th Symp. on Comp. Achitect., La Baule, France,1981

/KUH2/ _____
Some Extensions to the Theory of System Level Fault Diagnosis
Fault-Tolerant Computing, Symposium 10, Tokyo 1980

/KUH3/ _____
Fault Diagnosis in Fully Distributed Systems
Proc of the FTCS-11, 1981

/KUH4/ J.Kuhl
Fault Diagnosis in Computing Networks
PH.D.Thesis, University of Iowa, 1980

/KUN/ H.T.Kung, C.Leiserson
Systolic Arrays
Carnegie-Mellon University 1978
Research Review CMV-CS-79-103

/KUNG/ S.Y.Kung
VLSI Array Processor for Signal Processing
MIT Conf. on Advanced Res. on IC, Cambridge, MA 1980

/MAD/ R.F.Madden
An Algorithm for System Diagnosis
Science Institute, University of Iceland

/MAH/ S.Maheshwari, S.L.Hakimi
On Models for Diagnosable Systems and Probabilistic Fault Diagnosis
IEEE Transactions on Computers, Vol C25 March 1976

/MAEH/ E.Maehle, H.Joseph
Selbstdiagnose in fehlertoleranten Dirmu-
Multi-Mikroprozessorkonfigurationen
Fachtagung über Fehlertolerierende Rechnersysteme
IFB 54 Springer Verlag Berlin, 1982

/MAEH2/ E.Maehle
Fehlertolerante Rechnerstrukturen
Arbeitsberichte des Inst. f. Math. Maschinen,
Erlangen, Band 10/4

/MAE/ J.Maeng, M.Malek
A Comparison Connection Assignment for Self-Diagnosis
of Multiprocessor Systems
Proc. of the FTCS-11 1981

/MAL1/ M.Malek
A Comparison Connection Assignment
for Diagnosis of Multiprocessor Systems
Proc. of the 7th Symp. on Comp. Achitect., La Baule, France,1981

/MAL2/ M. Malek, J. Maeng
Partitioning of large Multicomputer Systems
for Efficient Fault-Diagnosis
IEEE Proc. of the FTCS-12

/MAN/ U.Manber
System Diagnosis with Repair
IEEE Transactions on Computers C29 Oct.1980

/MEA/ C.Mead, L.Conway
Introduction to VLSI-Systems
Addison-Wesley Publishing Company 1980

/MEY/ G.Meyer, G.Masson
An Efficient Fault Diagnosis Algorithm
for Symmetric Multiple Processor Architectures
IEEE Transactions on Computers Vol C27, Nov 1978

/MIE/ P.Mies, D.Schütt
Feldrechner
Bibliograph. Institut AG, Reihe Informatik 21, Mannheim 1976

/PEA1/ M.Pearse, R.Shostak, L.Lamport
Reaching Agreement in the Presence of Faults
Comm. of the ACM, Vol 27/2, April 1980

/PEA2/ M.Pearse, R.Shostak, L.Lamport
The Byzantine Generals Problem
ACM Tranact. on Progr. Lang. and Systems, Vol 4/3 J-ly 1982

/PHI/ Das N20DS Netzwerk
Philipps GmbH, Kassel

/PRA/ D.Pradhan, S.Reddy
A Fault-Tolerant Communication Architecture for Distributed Systems
Proc. of the FTCS-11, 1981

/PRE/ F.P.Preparata, G.Metze, R.T.Chien
On the Connection Assignment Problem of Diagnosable Systems
IEEE Transactions on Electronic Computers Vol EC-16 1967

/RIV/ R.Rivest, A.Shamir, L.Adleman
A Method for Obtaining Digital Signatures
and Public-Key Crypto Systems
CACM, Febr 1978, Vol 21/2

/ROB/ J.G.Robinson, E.Roberts
Software-Tolerance in the Pluribus-System
AFIPS Conference Proceedings Vol 47, Montvale, N.J.

/SAH/ F.Saheban, L.Simoncini, A.Friedman
Concurrent Computation and Diagnosis in Multiprocessor Systems
Proc. of the FTCS-9, 1981

/SAN/ N.Sandell, R.Tenney
Strategies for Distributed Decisionmaking
IEEE Transactions on Systems, Man and Cybernetics
Vol SMC-11/8 Aug. 1981

/SCH/ E.Schmitter et alii
Design eines fehlertoleranten Multimikrocmputersystems
BMFT- FB DV 80-005

/SEG/ A. Segall
Distributed Network Protocols
IEEE Trans. on Inform. Theory Vol IT 29/1, Jan 1981

/SMI/ J.E.Smith
Universal System Diagnosis Algorithm
IEEE Transactions on Computers Vol C28, May 1979

/SNY/ L.Snyder
Introduction to the Configurable, Highly Parallel Computer
IEEE Computer, Jan. 1982

/SON/ S.Song
A Highly Concurrent Tree Machine for Database Applications
Proc. of the Int. Conf. on Parallel Processing 1980
IEEE Cat.No 80CH 1569-3 VB

/STö/ H.Störmer
Mathematische Theorie der Zuverlässigkeit
Oldenbourg Verlag München 1970

/TEN/ R.Tenney, N.Sandell
Structures for Distributed Decision Making
IEEE Transactions on Systems, Man and Cybernetics
Vol SMC-11/8 Aug 1981

/TIL/ A. van Tilborg, L. Wittie
Wave Scheduling: Distributed Allocation of Task Forces
in Network Computers
Proc. of the 2. Int. Conf. on Distributed Comp. Systems, Paris 1981

/WEN/ Wensley et alii
SIFT: Design and Analysis of a Fault-Tolerant Computer
for Aircraft Control
Proc. of the IEEE, Vol 66/10, 1978

Lebenslauf

Am 27.8.1950 wurde ich, Rüdiger Brause, in Borna (DDR) als Sohn von Martin und Irmgard Brause geboren. Von Mai 1957 bis März 1963 besuchte ich die Helen-Keller Schule in Berlin-Schöneberg und ab April 1963 die Dreilinden-Oberschule in Berlin-Wannsee. Im Frühjahr 1970 legte ich dort das Abitur ab und immatrikulierte mich ab Sommersemester 1970 an der Universität des Saarlandes im Fach Physik, wo ich auch im Sommer 1973 die Diplomvorprüfung ablegte. Im Herbst 1973/74 wechselte ich zur Eberhard-Karls Universität Tübingen und legte dort im Juni 1978 das Physikdiplom ab. Das Thema meiner Diplomarbeit war 'Mustererkennung mit stochastischen Lernalgorithmen'. Während der folgenden zwei Jahre arbeitete ich im Forschungslabor der Nervenklinik Tübingen an dem Hardware- und Softwareaufbau einer Computersteuerung für psycho-physiologische Forschungsprogramme. Parallel dazu war ich am Institut für Informationsverarbeitung als wissenschaftliche Hilfskraft tätig und wurde dort ab September 1980 als wissenschaftlicher Angestellter im Rahmen des DFG- Projekts 'Technische Grundsatzfragen beim Einsatz von Mikroprozessoren' eingestellt. In dieser Zeit fertigte ich außerdem unter der wissenschaftlichen Anleitung von Prof. Dr. M. Dal Cin bis zum Herbst 1983 meine Dissertation über 'Selbstdiagnose von Mehrrechnersystemen bei nichtvollständiger Vernetzung' an.

Meine akademischen Lehrer waren vor anderen die Herren Professoren und Doktoren in Saarbrücken Lamprecht und Siems, in Tübingen Braitenberg, Dal Cin, Schmidt und Pfaffelhuber.

Danksagung

An dieser Stelle möchte ich mich bei Herrn Prof. Dr. M. Dal Cin für seine hilfreiche wissenschaftliche Betreuung bedanken. Weiterhin gilt mein besonderer Dank Herrn Dr. Ammann, dessen konstruktive Ideen viel zum 9.Kapitel beigetragen haben. Ebenso ist diese Arbeit an den fruchtbaren Diskussionen mit den Herren Dr. Dilger und Dr. Risse gewachsen.
Herrn Prof. Dr. W. Güttinger sei für die Bereitstellung der Arbeitsmittel am Institut für Informationsverarbeitung ebenfalls gedankt.
Schließlich möchte ich mich noch bei meiner Frau Nicole und meinem Sohn Patrick sehr herzlich bedanken, die geduldig und verständnisvoll Abende und Wochenenden auf mich verzichtet und damit zum Gelingen dieser Arbeit beigetragen haben.