(На правах рукописи)

АКАДЕМИЯ НАУК БССР

ОТДЕЛЕНИЕ ФИЗИКО-ТЕХНИЧЕСКИХ НАУК

Матросова Анжела Юрьевна

МЕТОДЫ РЕШЕНИЯ БУЛЕВЫХ УРАВНЕНИЙ И ИХ ПРИМЕНЕНИЕ В ДИАГНОСТИКЕ НЕИСПРАВНОСТЕЙ ДИСКРЕТНЫХ АВТОМАТОВ

(Специальность № 05.13.01 - "Техническая кибернетика и теория информации")

А в т о р е ф е р а т

диссертации на соискание ученой степени кандидата технических наук

Минск, 1973 г.

(На правах рукописи)

АКАДЕМИЯ НАУК БССР

ОТДЕЛЕНИЕ ФИЗИКО-ТЕХНИЧЕСКИХ НАУК

Матросова Анжела Юрьевна

МЕТОДЫ РЕШЕНИЯ БУЛЕВЫХ УРАВНЕНИЙ И ИХ ПРИМЕНЕНИЕ В ДИАГНОСТИКЕ НЕИСПРАВНОСТЕЙ ДИСКРЕТНЫХ АВТОМАТОВ

(Специальность № 05.13.01 - "Техническая кибернетика и теория информации")

А в т о р е ф е р а т

диссертации на соискание ученой степени кандидата технических наук

Минск, 1973 г.

В современной технике большую роль играют устройства по переработке дискретной информации. В качестве математической модели этих устройств принят дискретный автомат. Изучением дискретных автоматов занимается теория дискретных автоматов, основными вопросами которой являются задачи анализа и синтеза автоматов. Задачи анализа и синтеза разлагаются на более мелкие задачи, некоторые из них оказываются достаточно сложными в том смысле, что требуют для своего решения больших вычислительных затрат. В диссертационной работе рассматривается одна из таких задач, а именно, задача решения булева уравнения. Ей уделяется основное внимание: обсуждается ее связь с задачами теории дискретных автоматов, развиваются алгоритмы решения булевых уравнений, предлагается метод преобразования булевых уравнений, имеющий целью сократить вычислительные затраты на их решение. Кроме того, в реферируемой работе рассматриваются алгоритмы решения ряда задач анализа дискретных автоматов, возникающих в диагностике неисправностей автоматов и требующих многократного решения булевых уравнений. В приложении приводятся некоторые из разработанных автором программ, реализующих алгоиртмы, развитые в диссертационной работе. Программы составлены в языке ЛЯПАС.

Реферируемая работа состоит из введения, трех глав и приложения.

В о в в е д е н и и ставятся задачи, рассматриваемые в диссертационной работе, и разбирается вопрос о сведении задач теории дискретных автоматов к решению булевых уравнений.

Дано булево уравнение

$$F(x_1, \ldots, x_n) = \delta, \qquad (I)$$

где $\delta \in \{0,1\}$, $F(x_1, \ldots, x_n)$ - булева функция от аргументов $x_1, \ldots, x_n$. Назовем корнем уравнения (I) набор значений переменных $x_1, \ldots, x_n$, обращающий уравнение в тождество. Требуется найти все или некоторые корни уравнения, т.е. решить его.

Решить уравнение (I) можно путем перебора наборов значений переменных $x_1, \ldots, x_n$ и вычисления на каждом из них

значения функции $f(x_1, \ldots, x_n)$. Выполнение такой процедуры требует больших вычислительных затрат при больших значениях n аргументов функции. В связи с этим целесообразно рассматривать некоторые классы уравнений (I) и развивать методы решения уравнений, учитывающие специфику классов. В этом направлении выполнен ряд работ [I-7]. В них рассматриваются в основном такие уравнения (I), в которых функция $f(x_1, \ldots, x_n)$ представляется в виде логического произведения нескольких булевых функций:

$$f(x_1, \ldots, x_n) = f_1(x_1, \ldots, x_n) \wedge \ldots \wedge f_k(x_1, \ldots, x_n). \qquad (2)$$

Авторы работ [I,3] предлагают методы решения уравнений, в которых каждая функция $f_i(x_1, \ldots, x_n)$ существенно зависит лишь от некоторых аргументов из $\{x_1, \ldots, x_n\}$, $i = 1, \ldots, k$. В работах [4,6,7] развиваются процедуры решения уравнений, ориентированные на представление функции $f_i(x_1, \ldots, x_n)$ в виде д.н.ф., причем, $\delta = 1$.

В диссертационной работе предлагаются методы решения булевых уравнений более общего вида. В них функция $f(x_1, \ldots, x_n)$ задана в так называемой скобочной форме.

Под скобочной формой будем понимать следующее. Назовем (элементарными) скобочными формами выражения $x_1, \ldots, x_n, 0, 1$. Если $A_1, \ldots, A_\ell, \ldots, A_k$ – скобочные формы, то выражения $(A_1 \vee \ldots \vee A_k)$, $(A_1 \wedge \ldots \wedge A_k)$, $\bar{A}_\ell$ – тоже скобочные формы.

Методы, развитые в работах [I-7], как правило, не могут быть использованы для решения уравнений, в которых функция $f(x_1, \ldots, x_n)$ представлена в скобочной форме. Для них нужны специальные алгоритмы. Описанию таких алгоритмов посвящена первая глава диссертационной работы.

Диагностика неисправностей дискретных автоматов в настоящее время является интенсивно развивающимся направлением теории дискретных автоматов. Это направление стимулирует интерес ко многим задачам анализа автоматов. Одной из них является анализ комбинационной сети с целью построения теста для данной неисправности. Эта задача может быть сведена к решению уравнения (I), левая часть которого является скобочной формой. В работах [8-10] к решению уравнения такого вида предлагается переходить не сразу, а лишь после выполнения некоторых процедур, позволяющих в конечном счете иметь дело с более простыми уравнениями. И все-таки вычислительные трудности, связанные с решением этих уравнений, могут оказаться весьма значительными. Вот почему представляет интерес рассмотрение комбинационных сетей некоторых частных классов и развитие специально для них методов построения теста, позволяющих, если не обойти, то существенно уменьшить эти трудности. Такой подход отражен во второй главе диссертационной работы. В ней рассматриваются алгоритмы построения теста для данной неисправности в k-ярусной комбинационной сети.

При диагностике неисправностей в автоматах с памятью, в частности, в асинхронных автоматах, необходимо знать область работы автомата, в которой его поведение оказывается предсказуемым. Знание такой области важно, например, при построении последовательности входных воздействий, позволяющей по соответствующей последовательности реакций отличить исправный автомат от автомата с данной неисправностью.

Результаты работы [II] могут рассматриваться в качестве одного из возможных подходов к отысканию такой области. В работе [II] предложена процедура, позволяющая выяснить, как будет вести себя автомат, находящийся в заданном состоянии, при поступлении на него некоторого входного сигнала. Задача решается в довольно общем виде: состояния могут быть неполностью определенными, а соотношения скоростей изменения сигналов в отвечающем автомату реальном устройстве – произвольными. Часто в нашем распоряжении имеется некоторая информация о реальном устройстве, представляемом асинхронным автоматом. Например, нам известно, что следует избегать лишь опасных состязаний между элементами памяти. В этом случае возникает задача анализа автомата на выявление в нем области работы, в которой отсутствуют опасные состязания. Эта задача рассматривается в третьей главе диссертации.

Первая глава посвящена решению булевых уравнений. В начале главы (§ I) приводится обзор известных методов решения. Затем развиваются алгоритмы решения булевых уравнений (I), в которых левая часть представлена в скобочной форме. Предварительно (в § 2) вводятся необходимые определения.

Выражение, заключенное в некоторые скобки (...), и любую часть этого выражения будем называть подчиненными этим скобкам. Скобки назовем подчиняющими, а отношение между скобками и выражением - отношением подчинения.

Назовем глубиной скобок (...) число других скобок, которым рассматриваемые скобки подчинены. Скобки, глубины которых отличаются на единицу, будем называть соседними по глубине. Если некоторый символ из $\{\wedge, \vee, x_1, \ldots, x_n\}$ подчинен скобкам i-ой глубины и не подчинен скобкам $(i+1)$-ой глубины, то будем считать, что этот символ входит в скобки i-ой глубины свободным образом.

Если один или несколько символов дизъюнкции (конъюнкции) входят свободным образом в некоторые скобки i-ой глубины, то будем говорить, что операцией рассматриваемых скобок является дизъюнкция (конъюнкция).

Число операндов, к которым относится операция некоторых скобок, назовем характеристикой этих скобок.

Скобочную форму будем называть упорядоченной, если для нее выполняются следующие условия:

I. Знаки отрицания стоят только над символами переменных.

2. Всем скобкам одинаковой глубины соответствует одна и та же операция из $\{\wedge, \vee\}$.

3. Соседним по глубине скобкам соответствуют двойственные операции из $\{\wedge, \vee\}$.

От скобочной формы $A(x_1, \ldots, x_n)$ произвольного вида легко перейти к упорядоченной скобочной форме $A^*(x_1, \ldots, x_n)$.

Примером упорядоченной скобочной формы может служить выражение

$$((((x_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_3)) \vee ((x_1 \vee \bar{x}_3) \wedge (x_2 \vee \bar{x}_4))) \wedge \quad (3)$$
$$\wedge ((x_1 \wedge x_2) \vee (x_3 \wedge x_4 \wedge \bar{x}_2 \wedge x_1))).$$

Структуру выражения $A^*(x_1, \ldots, x_n)$ можно представить деревом $\mathcal{G}^*$. В нем каждым скобкам из $A^*(x_1, \ldots, x_n)$ сопоставляется неконцевая вершина, каждому символу переменной - концевая вершина, а каждому отношению подчинения между соседними по глубине скобками или отношению подчинения между символом переменной и скобками, в которые он входит свободным образом - ребро.

Вершину, соответствующую скобке нулевой глубины, назовем корнем дерева.

Отметим концевые вершины дерева символами соответствующих им переменных, а неконцевые вершины - символами операций скобок.

Договоримся строить дерево так, что если в выражении $A^*(x_1, \ldots, x_n)$ некоторые скобки записаны левее других скобок, то поддерево, соответствующее выражению в первых скобках, лежит левее поддерева, соответствующего выражению во вторых скобках. Будем считать, что символы переменных в выражении $A^*(x_1, \ldots, x_n)$ пронумерованы слева направо, тогда в дереве $\mathcal{G}^*$ j-ому символу переменной выражения $A^*(x_1, \ldots, x_n)$ соответствует j-ая слева концевая вершина.

Структура упорядоченной скобочной формы (3) представляется деревом рис.I.

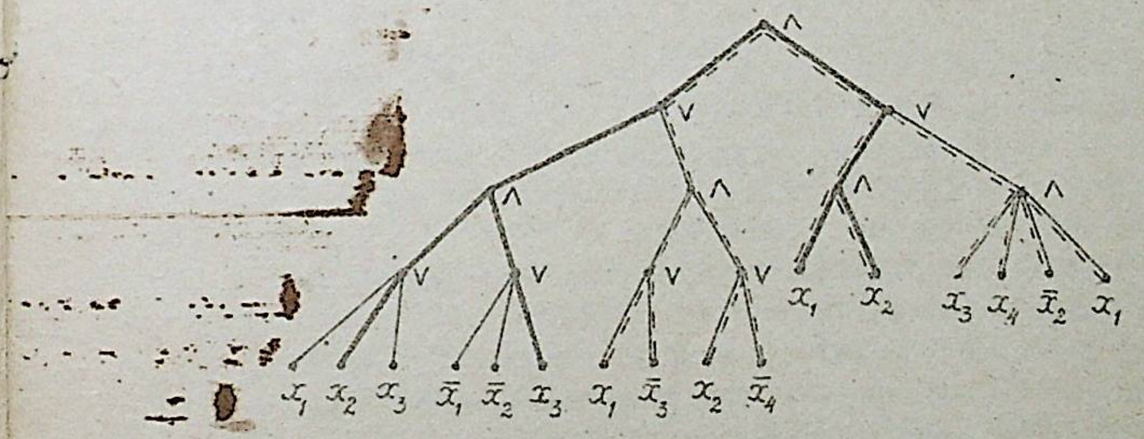

Рис. 1.

Дерево $\overline{\mathcal{G}^*}$, задающее структуру скобочной формы $\overline{\mathcal{A}^*}$, получается из дерева $\mathcal{G}^*$ путем замены операций на двойственные, а переменных – на инверсные.

Далее в § 2 рассматриваются эквивалентные преобразования дерева $\mathcal{G}^*$, выполняемые с целью сокращения числа его концевых вершин.

В § 3 приводится алгоритм отыскания всех корней булева уравнения:

$$\mathcal{A}^*(x_1,\dots,x_n)=\delta,\quad \delta\in\{0,1\}, \tag{4}$$

где $\mathcal{A}^*(x_1,\dots,x_n)$ – упорядоченная скобочная форма.

Отыскание всех корней этого уравнения можно свести к последовательному раскрытию скобок (с целью получения д.н.ф.) в выражении $\mathcal{A}^*(x_1,\dots,x_n)$, если $\delta=1$, или $\overline{\mathcal{A}^*}(x_1,\dots,x_n)$, если $\delta=0$. Операция раскрытия скобок, как правило, сопровождается "расширением" исходного выражения, тем бо́льшим, чем больше число n переменных.

В реферируемой работе предлагается алгоритм отыскания всех корней уравнения

$$\mathcal{A}^*(x_1,\dots,x_n)=1, \tag{5}$$

не требующий расширения исходного выражения и сводящийся к перебору некоторых из так называемых существенных поддеревьев дерева $\mathcal{G}^*$, сопоставляемого выражению $\mathcal{A}^*(x_1,\dots,x_n)$.

Назовем существенным поддеревом дерева $\mathcal{G}^*$ такое поддерево, в которое

1) всегда входит корень дерева $\mathcal{G}^*$,

2) локальные степени вершин, отмеченных символом $\wedge$, совпадают с локальными степенями тех же вершин дерева $\mathcal{G}^*$,

3) локальная степень корня поддерева, отмеченного символом $\vee$, равна единице, а локальные степени остальных вершин, отмеченных символом $\vee$, равны двум.

В дереве рис.1 одно из существенных поддеревьев выделено жирными линиями.

Две концевые вершины в дереве $\mathcal{G}^*$ назовем взаимно инверсными, если они отмечены символами одной и той же переменной, но с разными знаками.

Если среди концевых вершин, принадлежащих существенному поддереву, нет взаимно инверсных, то ему может быть

сопоставлена конъюнкция γ, образованная из символов переменных, отмечающих концевые вершины. Будем говорить в этом случае, что существенное поддерево порождает конъюнкцию γ.

Т е о р е м а 1. Конъюнкции, порождаемые существенными поддеревьями дерева $\mathcal{G}^*$, представляют все корни уравнения (5).

Из теоремы 1 следует, что нахождение всех корней уравнения (5) можно свести к перебору всевозможных существенных поддеревьев.

Перебор сокращается за счет отбрасывания существенных поддеревьев, заведомо не порождающих конъюнкций.

Выявление таких поддеревьев можно осуществить, если строить очередное существенное поддерево, начиная с концевых вершин. Как только среди концевых вершин оказываются взаимно инверсные, поддерево не достраивается, а вместе с ним в общем случае не достраиваются целые совокупности существенных поддеревьев.

Если достаточно найти один (любой) корень уравнения (5), то следует ограничиться отысканием одного существенного поддерева, порождающего конъюнкцию γ.

При решении уравнения

$$\mathcal{A}^*(x_1,\dots,x_n)=0$$

необходимо перебрать существенные поддеревья дерева $\overline{\mathcal{G}^*}$.

В § 4 первой главы приводится метод, специально ориентированный на нахождение одного (любого) корня уравнения (5). Он основан на применении следующего утверждения.

Т е о р е м а 2. Если в уравнении $\mathcal{A}^*(x_1,\dots,x_n)=1$ переменная x_i встречается только с инверсией (только без инверсии), то среди корней этого уравнения, если они существуют, найдется по крайней мере один, в котором x_i принимает нулевое (единичное) значение.

Подобное утверждение было сформулировано в работе [5] для уравнения $\mathcal{F}(x_1,\dots,x_n)=0$, в котором $\mathcal{F}(x_1,\dots,x_n)$ задана в д.н.ф.

Теорема 2 применяется как к исходному выражению $\mathcal{A}^*(x_1,\dots,x_n)$, так и к его остаткам с целью фиксиро-

вания значений соответствующих переменных. Остатком выражения $A^*(x_1,\dots,x_n)$ называется новое выражение, полученное из $A^*(x_1,\dots,x_n)$ путем замены константами символов некоторых переменных. Если условия теоремы 2 не выполняются, то фиксируется значение одной переменной, например, такой, которая встречается в $A^*(x_1,\dots,x_n)$ или в остатке наиболее часто.

Переменные, значения которых зафиксированы по теореме 2, образуют множество неотмеченных переменных. Остальные зафиксированные переменные образуют множество отмеченных переменных. При отыскании одного (любого) корня уравнения (5) организуется перебор по отмеченным переменным. Алгоритм аналогичен алгоритму, изложенному в [5].

Эксперименты на ЭВМ показали, что рассматриваемый метод отыскания одного (любого) корня уравнения (5) оказывается эффективным, если каждая переменная встречается в выражении $A^*(x_1,\dots,x_n)$ в среднем редко, не более 4-х - 5-и раз.

В конце § 4 первой главы приводится процедура, позволяющая продвинуться в область более частых вхождений переменных.

Решение уравнения (5) сводится к решению методом перебора по отмеченным переменным нескольких в некотором смысле более "простых" уравнений того же вида. Любой корень "простого" уравнения является в то же время корнем исходного уравнения. Уравнение считается "простым", если оно решается достаточно быстро методом перебора по отмеченным переменным. Левая часть каждого "простого" уравнения может быть представлена так называемым порождающим поддеревом дерева $\mathcal{Y}^*$.

Порождающим поддеревом дерева $\mathcal{Y}^*$ называется такое поддерево, в которое

1) всегда входит корень дерева $\mathcal{Y}^*$,

2) локальные степени вершин, отмеченных символом $\wedge$, совпадают с локальными степенями тех же вершин дерева $\mathcal{Y}^*$,

3) локальная степень корня поддерева, отмеченного символом $\vee$, ≥ 1, а локальные степени остальных отмеченных символом $\vee$ вершин ≥ 2.

Одно из порождающих поддеревьев выделено на рис.1 пунктирными линиями.

Предлагается выполнять перебор порождающих поддеревьев дерева $\mathcal{Y}^*$, придерживаясь следующих правил:

1. Пусть q - среднее число вхождений переменной, при котором уравнение (5) решается достаточно быстро методом перебора по отмеченным переменным, а ρ - среднее число вхождений переменной в некоторое порождающее поддерево. Рассматриваются только поддеревья, для которых $\rho \leq q$.

2. Перебираются возможно более крупные поддеревья, т.е. поддеревья с бо́льшим числом концевых вершин.

В § 5 той же главы приводится процедура преобразования булева уравнения (1), в котором функция $F(x_1,\dots,x_n)$ представлена в виде (2), причем, $f_i(x_1,\dots,x_n)$ заданы в д.н.ф., а $\delta = 1$, $i = 1,\dots,\kappa$. Преобразование выполняется с целью сокращения объема вычислений при решении уравнения и таким образом, что корни преобразованного уравнения являются корнями исходного уравнения и наоборот. В основе алгоритма преобразования лежит идея выбрасывания корней уравнений

$$f_i(x_1,\dots,x_n) = 1, \qquad (6)$$

не являющихся корнями заданного уравнения. Выбрасывание корней выполняется путем удаления из уравнений (6) некоторых конъюнкций, а так же путем замены некоторых конъюнкций конъюнкциями бо́льшего ранга, имплицирующими заменяемые.

Методы решения булевых уравнений были представлены в виде программ на языке ЛЯПАС и испытывались на ЭВМ (М-220). Результаты испытаний приводятся в § 6 первой главы реферируемой работы. Н.Г.Дроновой под руководством автора был составлен генератор выражений $A^*(x_1,\dots,x_n)$. Исходными данными для генератора являлись: число n переменных, число l символов переменных (концевых вершин в $\mathcal{Y}^*$) и интервал κ характеристик, из которого в процессе построения выражения $A^*(x_1,\dots,x_n)$ выбирается случайным образом характеристика для очередной строящейся скобки.

Отыскивались все корни уравнения (5) методом сокращенного перебора существенных поддеревьев. Результаты экспе-

римента показали, что затраты времени на отыскание всех решений сильно возрастают с увеличением числа ℓ и "расширением" интервала κ характеристик. Это объясняется резким увеличением общего числа существенных поддеревьев. Так, при $\ell = 200_8$, $n = 14_8$, $\kappa = (2)$, $t = 4''$, а при $\ell = 400_8$ и тех же значениях n и κ $t = 2'5''$. При $\ell = 200_8$, $n = 14_8$, $\kappa = (2,3,4,5)$ уравнение (5) решалось 12 минут, были найдены 4000 его корней, а перебор существенных поддеревьев не закончился. При $\ell = 300_8$, $\kappa = (2)$ и изменении значений n от 14_8 до 200_8 время отыскания всех корней уравнений не превышало $3'5''$.

Методом сокращенного перебора существенных поддеревьев отыскивался один корень уравнения (5). При $\ell = 1000_8$, $\kappa = (2)$ и изменении значений n от 14_8 до 400_8 время отыскания одного корня не превышало 4-х минут. Для $\ell = 400_8$ $\kappa = (2,3,4,5)$ и значений n, меняющихся от 14_8 до 200_8, время отыскания одного корня не превышало 13 секунд.

Испытывался так же метод перебора по отмеченным переменным. Решались уравнения для $\ell = 200_8$, 400_8, 1000_8, $n = 20_8$, 40_8, 60_8, 100_8, 200_8, $\kappa = (2,3)$ (около сотни уравнений). Корень, как правило, отыскивался за несколько (до 10-15) секунд. Это объясняется тем, что генерируемые уравнения имели много корней.

Чтобы выяснить, как будут решаться уравнения, имеющие мало корней, определялось, какая часть λ переменных из зафиксированных при отыскании корня оказывается отмеченной. По этим переменным пришлось бы выполнять перебор, если бы корень быстро не отыскался. Оказалось, что при средней частоте вхождения переменной в $A^*(x_1, \ldots, x_n)$, близкой к четырем, значение λ менялось в пределах от 0,04 до 0,1, т.е. в худшем случае пришлось бы выполнять перебор по $1/10$ переменных от общего числа переменных.

При отыскании одного (любого) корня уравнения (5) методом перебора порождающих поддеревьев значение q выбиралось равным четырем. Корень обычно отыскивался в несколько раз быстрее, чем при применении алгоритма перебора по отмеченным переменным непосредственно к выражению $A^*(x_1, \ldots, x_n)$.

В о в т о р о й г л а в е рассматривается задача построения теста для данной неисправности в комбинационной сети.

Сопоставим входам комбинационной сети переменные $x_1, \ldots, x_n$, а выходам сети - переменные $y_1, \ldots, y_m$. Назовем набор значений переменных $x_1, \ldots, x_n$ входным воздействием сети, а соответствующий ему набор значений переменных $y_1, \ldots, y_m$ - реакцией сети.

Будем считать, что сеть не исправна, если некоторые ее элементы реализуют булевы функции, отличные от функций, реализуемых ими в исправном состоянии сети.

Тестом для данной неисправности назовем входное воздействие, при котором реакция исправной сети отлична от реакции сети в присутствии данной неисправности. Будем говорить, что тест обнаруживает эту неисправность.

Проверяющим тестом для множества γ неисправностей в комбинационной сети назовем совокупность входных воздействий, в которой найдется хотя бы один тест для любой неисправности из γ.

Проверяющий тест можно строить следующим образом.

1. Найти по одному тесту для каждой неисправности из γ, а затем тесты объединить.

2. Найти тест для некоторой неисправности из γ. Выяснить, для каких других неисправностей множества γ он одновременно является тестом. Затем найти тест для одной из не обнаруживаемых первым тестом неисправностей и аналогичным образом анализировать его и т.д., до тех пор, пока не будет построен проверяющий тест.

Подходы 1,2, хотя и не гарантируют построения минимального проверяющего теста, привлекательны своей простотой. В них можно выделить задачу построения одного (любого) теста для данной неисправности.

В § 2 второй главы показывается, как можно свести задачу построения теста для данной неисправности в сети κ решению булева уравнения (5). Сеть представляется графом соединений и списком функций, реализуемых элементами.

В § 3 второй главы рассматривается задача построения теста для данной неисправности в κ-ярусной комбинацион-

ной сети [7].

κ-ярусной комбинационной сетью называется сеть, в которой все элементы могут быть разбиты на классы (ярусы) с номерами $1, \ldots, \kappa$, так чтобы любая межэлементная связь связывала выходной полюс некоторого элемента i-го яруса с входным полюсом элемента $(i+1)$-го яруса, $i = 1, \ldots, \kappa-1$. Кроме того, входные полюсы такой сети могут непосредственно связываться лишь с входными полюсами элементов первого яруса, а выходными полюсами сети могут служить лишь выходные полюсы элементов последнего κ-го яруса.

В основе предлагаемых алгоритмов нахождения теста для данной (одиночной) неисправности в κ-ярусной сети лежит идея построения так называемой таблицы реакций.

Таблица строится один раз, а затем многократно анализируется с целью отыскания тестов для различных (одиночных) неисправностей.

Для сетей с небольшим (порядка десяти) числом входов строится полная таблица реакций из κ столбцов и r строк, где κ -число ярусов сети, а $r = 2^n$ - число всевозможных входных воздействий, подаваемых на сеть. На пересечении j-го столбца и i-ой строки записывается реакция j-го яруса сети на i-ое входное воздействие,

$j \in \{1, \ldots, \kappa\}$, $i \in \{1, \ldots, 2^n\}$.

Для сетей с несколькими десятками входов предлагается строить сокращенную таблицу реакций из κ столбцов и t строк, где t -число различных реакций первого яруса сети. Построение первого столбца таблицы реакций связано с решением булевых уравнений вида

$$F(x_1, \ldots, x_n) = 0, \qquad (8)$$

в которых $F(x_1, \ldots, x_n)$ представляется в д.н.ф.

Анализ полной таблицы реакций представляет собой процедуру слежения за распространением последствий неисправности до выходов схемы. Анализ сокращенной таблицы реакций, кроме процедуры слежения требует для некоторых неисправностей решения булевых уравнений (8).

Чем большее число N неисправностей, для которых ищется тест, тем выгоднее пользоваться предложенными алгоритмами. Дело в том, что время, затрачиваемое на построение теста для одной неисправности, складывается из времени построения таблицы, поделенного на N, и времени анализа таблицы.

В § 3 также обсуждается возможность сведения произвольных комбинационных сетей к κ-ярусным.

В конце § 3 приводятся результаты испытаний алгоритмов построения теста для данной неисправности в κ-ярусной сети на ЭВМ (М-220). Были составлены две программы, реализующие построение теста для одиночной неисправности. Одна предназначена для сетей с числом входов не больше 10, а другая - для сетей с двумя-тремя десятками входов.

При проведении эксперимента использовался генератор κ-ярусных сетей, составленный Ю.Д.Черкашиным. Для схем, число входов которых не превышает 10, полные таблицы реакций строились не более 30 секунд.

Время, затрачиваемое на построение сокращенной таблицы реакций существенно зависит от числа элементов в первом ярусе сети. Так при $q = 8$ время построения таблицы составляет 35 секунд, а при $q = 12$ уже 7 минут. В обоих случаях элементы сети реализуют функции типа "и", "или", "не-и", "не-или", "не". Затраты времени на построение сокращенной таблицы реакций возрастают примерно в два раза с увеличением числа элементов в первом ярусе на единицу. Рост числа элементов сети при неизменном числе элементов в первом ярусе существенно не сказывается на времени построения сокращенной таблицы реакций. Анализировались таблицы реакций, построенные для сетей с 10-ю, 20-ю, 30-ю входами, $q = 8$. Элементам сети сопоставлялись различные булевы функции: а) д.н.ф., б) "не и", в) "не или", г) "не-и", "и", "не-или", "или", "не". Было построено около 160 тестов. Время анализа таблицы реакций для одного теста составляет 1-3 секунды.

Возможности метода построения теста для κ-ярусной сети с несколькими десятками входов ограничены числом q

элементов в первом ярусе. Для больших значений q таблицу реакций нельзя представить в оперативной памяти машины, а использование внешней памяти увеличит затраты времени на реализацию метода.

В третьей главе рассматриваются задачи анализа асинхронных автоматов на отсутствие опасных состязаний между элементами памяти.

Явление состязания между элементами памяти в автомате обусловлено инерционностью и нестандартностью элементов памяти в соответствующем реальном устройстве. Если в результате состязания автомат в конце концов приходит в предписанное ему состояние, то такое состязание считают неопасным, в противном случае состязание называют опасным.

Предполагается, что асинхронный автомат задается системой булевых уравнений:

$$z_i' = f_i(x_1,\dots,x_n,z_1,\dots,z_\rho) \qquad (9)$$

$$i = 1,\dots,\rho$$

$$y_j = g_j(x_1,\dots,x_n,z_1,\dots,z_\rho)$$

$$j = 1,\dots,m,$$

в которой $x_1,\dots,x_n$; $z_1,\dots,z_\rho$; $y_1,\dots,y_m$ - входные, внутренние и выходные переменные, а $f_i(x_1,\dots,x_n,z_1,\dots,z_\rho)$, $g_j(x_1,\dots,x_n,z_1,\dots,z_\rho)$ - функции переходов и выходов автомата. Значение переменной z_i' относится к последующему моменту времени по сравнению со значением переменной z_i.

В § I третьей главы предлагаются две процедуры построения таблиц переходов асинхронного автомата по заданным уравнениям (9). Во всех переходах таблиц отсутствуют опасные состязания между элементами памяти. Состояния, представляемые элементами таблиц, задаются наборами значений булевых переменных. Методы построения таблиц переходов сводятся к выполнению следующих операций.

I. Находится множество устойчивых состояний.

2. Для каждого устойчивого состояния u отыскивается множество полных неустойчивых состояний $Q(u)$ специального вида. Из каждого состояния множества осуществляется переход без опасных состязаний в устойчивое состояние u.

3. По найденным множествам $Q(u)$ и множеству устойчивых состояний строится таблица переходов.

Методы построения таблиц переходов ориентированы на автоматы, у которых число устойчивых состояний не особенно велико (около сотни), они позволяют строить таблицы переходов в общем случае существенно быстрее, чем путем перебора всевозможных пар устойчивых состояний. Рассматриваются множества $Q(u)$ двух типов.

К первому типу относятся множества, в которых из каждого состояния $q \in Q(u)$ в автомате реализуется единственная последовательность элементарных переходов

$q = e_1 \to e_2, e_2 \to e_3, \dots, e_{t-1} \to e_t = u$, $t = 1,\dots,r$, $r < 2^\rho$. Каждый элементарный переход $e_j \to e_{j+1}$ есть переход между соседними состояниями, $j = 1,\dots,t-1$. Два состояния называются соседними, если представляющие их наборы значений переменных отличаются значением одной переменной из $\{z_1,\dots,z_\rho\}$. Переход $q \to u$ в этом случае будем называть переходом первого типа.

Ко второму типу относятся множества, в которых из каждого состояния $q \in Q(u)$ выполняется прямой переход в u.

Обозначим наборы значений переменных, представляющие состояния q, u через $x_1^q,\dots,x_n^q, z_1^q,\dots,z_\rho^q$; $x_1^u,\dots,x_n^u, z_1^u,\dots,z_\rho^u$.

Пусть
$$z_i^u = f_i(x_1^q,\dots,x_n^q,z_1^q,\dots,z_\rho^q) \qquad (10)$$
$$i = 1,\dots,\rho$$
$$x_j^u = x_j^q$$
$$j = 1,\dots,n.$$

Обозначим через α минимальный подкуб единичного $(n+\rho)$-мерного куба, включающий $x_1^q,\dots,x_n^q, z_1^q,\dots,z_\rho^q$; $x_1^u,\dots,x_n^u, z_1^u,\dots,z_\rho^u$. Переход $q \to u$ прямой, если все наборы значений переменных, соответствующие элементам из α, являются корнями системы (10).

В § I развивается метод построения таблиц переходов,

в которых все переходы принадлежат первому типу, а затем предлагается метод построения таблиц, в которых все переходы прямые.

Процедуры построения множества устойчивых состояний и множеств $Q(u)$ обоих типов связаны с решением булевых уравнений (I), в которых $\mathcal{F}(x_1, \ldots, x_n)$ задана в виде (2), причем $f_i(x_1, \ldots, x_n)$ представлены в д.н.ф., а $\delta = 1, i = 0\,1, \ldots, \kappa$.

Пусть число устойчивых состояний в асинхронном автомате велико (порядка тысяч), и таблица переходов для него практически не может быть построена. Автомат задан системой (9). В этом случае предлагается строить множество $\mathcal{K}$ недопустимых состояний, такое что его дополнение $\overline{\mathcal{K}}$ задает область работы автомата, в которой отсутствуют опасные состязания между элементами памяти. $\overline{\mathcal{K}}$ включает в себя все устойчивые состояния автомата и все неустойчивые состояния, из которых выполняются прямые переходы в устойчивые состояния.

Назовем недопустимым всякое неустойчивое состояние q, из которого не реализуется прямой переход в состояние α, определяемое соотношениями (IO).

Если q – недопустимое состояние, то найдется такое состояние $z \in \alpha$ и такое $i \in \{1, \ldots, \rho\}$, что

$$z_i^u \neq f_i(x_1^z, \ldots, x_n^z, z_1^z, \ldots, z_\rho^z). \qquad (II)$$

Множество $\mathcal{K}$ недопустимых состояний строится по заданным уравнениям (9). Предполагается, что наряду с функциями $f_i(x_1, \ldots, x_n, z_1, \ldots, z_\rho)$ в нашем распоряжении имеются функции $\bar{f}_i(x_1, \ldots, x_n, z_1, \ldots, z_\rho)$. Множество $\mathcal{K}$ представляется булевой функцией $\psi(x_1, \ldots, x_n, z_1, \ldots, z_\rho)$, такой что наборы значений переменных, на которых $\psi(x_1, \ldots, x_n, z_1, \ldots, z_\rho)$ принимает единичное значение, задают недопустимые состояния.

Множество $\mathcal{K}$ может быть построено путем перебора полных состояний и проверки для каждого из них условий (II). Состояния, для которых выполняются условия (II), включаются в $\mathcal{K}$. Выполнить такую процедуру для достаточно больших n и ρ практически невозможно, даже используя ЭВМ.

В реферируемой работе предлагается иной алгоритм отыскания множества недопустимых состояний. Он сводится к перебору некоторых подмножеств букв конъюнкций д.н.ф., представляющих $f_i(x_1, \ldots, x_n, z_1, \ldots, z_\rho)$, $\bar{f}_i(x_1, \ldots, x_n, z_1, \ldots, z_\rho)$, и к решению булевых уравнений (I), соответствующих некоторым из перебираемых подмножеств букв. Левая часть уравнений (I) представлена в виде (2), причем, $f_i(x_1, \ldots, x_n)$ заданы в д.н.ф., а $\delta = 1$. В § 2 доказываются три теоремы, на основании которых строится алгоритм отыскания множества недопустимых состояний.

Обозначим через t некоторое подмножество внутренних переменных, $t \subseteq \{z_1, \ldots, z_\rho\}$. Для функции $f_i(x_1, \ldots, x_n, z_1, \ldots, z_\rho)$, $i \in \{1, \ldots, \rho\}$ и для t определим множество $\mathcal{K}_{i,t}$:

$$q \in \mathcal{K}_{i,t} \leftrightarrow [z_i^u = f_i(q) \neq f_i(q_t)] \; (q_t \in \alpha) \qquad (I2)$$

Здесь q_t отличается от q значениями всех переменных множества t и только ими;

z_i^u – значение переменной z_i в наборе $x_1^u, \ldots, x_n^u, z_1^u, \ldots, z_\rho^u$, определяемом соотношением (IO);

α – минимальный подкуб единичного $(n+\rho)$-мерного куба, включающий элементы $x_1^q, \ldots, x_n^q, z_1^q, \ldots, z_\rho^q$; $x_1^u, \ldots, x_n^u, z_1^u, \ldots, z_\rho^u$.

Будем выбирать всевозможные подмножества s из $\{z_1, \ldots, z_\rho\}$ и строить для каждого t и каждой функции $f_i(x_1, \ldots, x_n, z_1, \ldots, z_\rho)$ множество $\mathcal{K}_{i,t}$. Объединив все такие множества, найдем множество $\mathcal{K}$ недопустимых состояний,

$$\mathcal{K} = \bigcup_t \bigcup_{i=1}^{\rho} \mathcal{K}_{i,t}.$$

Обозначим через s любое из непустых подмножеств множества t, не совпадающее с t, $s \subset t$. Пусть q_s отличается от q значениями переменных множества s и только ими, а $\mathcal{K}_{i,s}$ аналогично множеству $\mathcal{K}_{i,t}$.

Теорема 3. Если $q \in \mathcal{K}_{i,t}$,
то $q_s \in \alpha$.

Теорема 4. Если 1. $q \in \mathcal{K}_{i,t}$,
2. $f_i(q_t) = f_i(q_s)$,

то $q \in \mathcal{K}_{i,s}$.

Следствие. Если I. $q \in \mathcal{K}_{i,t}$,

2. $q \notin \mathcal{K}_{i,s}$,

то $f_i(q_t) \neq f_i(q_s)$.

Обозначим символом β множество конъюнкций в д.н.ф., представляющих функции

$$f_j(x_1,\dots,x_n,z_1,\dots,z_\rho),\quad \overline{f}_j(x_1,\dots,x_n,z_1,\dots,z_\rho),$$
$$j=1,\dots,\rho,$$

символом β_i, $\beta_i \subseteq \beta$, множество конъюнкций д.н.ф. двух функций $f_i(x_1,\dots,x_n,z_1,\dots,z_\rho)$ и $\overline{f}_i(x_1,\dots,x_n,z_1,\dots,z_\rho)$, а символом γ некоторую конъюнкцию множества β, $\gamma \in \beta$. Под символом t_γ будем понимать множество внутренних переменных, являющихся буквами конъюнкции γ. Обозначим через α_γ множество полных состояний, соответствующих конъюнкции γ.

Теорема 5. Если: I. $q \in \mathcal{K}_{i,t}$,

2. $q \notin \mathcal{K}_{i,s}$ для всех $s \subset t$,

то найдется такая конъюнкция $\gamma \in \beta_i$, что:

I. $q_t \in \alpha_\gamma$,

2. $t \subseteq t_\gamma$.

Обозначим через $\mathcal{K}^*_{i,t}$ подмножество множества $\mathcal{K}_{i,t}$ такое, что

$$\mathcal{K}^*_{i,t} = \mathcal{K}_{i,t} \setminus \bigcup_{s \subset t} \mathcal{K}_{i,s}.$$

Следствие. Если среди конъюнкций множества β_i не найдется ни одной конъюнкции γ, такой, что $t \subseteq t_\gamma$, то $\mathcal{K}^*_{i,t} = \phi$.

Множество недопустимых состояний может быть представлено в виде

$$\mathcal{K} = \bigcup_t \bigcup_{i=1}^{\rho} \mathcal{K}^*_{i,t}.$$

Процедура построения множества $\mathcal{K}$ сводится к отысканию всевозможных $\mathcal{K}^*_{i,t}$.

В конце § 2 приводятся результаты испытаний программы, реализующей алгоритм. Строились множества недопустимых состояний для 4 реально действующих автоматов. Число входных переменных в автоматах равнялось 5-6, число внутренних переменных 8-I2. Худшее время, затрачиваемое на построение множества, составляло 2'5", лучшее - I5". Максимальное число конъюнкций функции γ равнялось I47, а минимальное - 73.

В реферируемой работе получены следующие основные результаты:

I. Показана связь задачи отыскания корней булевых уравнений с различными задачами теории дискретных автоматов.

2. Разработан ряд методов решения булевых уравнений, заданных в так называемой скобочной форме.

а) Метод сокращенного перебора существенных поддеревьев, позволяющий находить все корни уравнения.

б) Два метода решения булевых уравнений, ориентированные на отыскание одного (любого) корня уравнения.

3. Указаны возможности использования развитых в работе методов решения булевых уравнений в задаче построения теста для данной неисправности в комбинационной сети.

4. Разработаны алгоритмы решения ряда задач анализа дискретных автоматов, возникающих в диагностике неисправностей автоматов. Алгоритмы связаны с решением булевых уравнений.

а) Предложены два метода построения теста для данной неисправности в κ-ярусной сети.

б) Развиты две процедуры построения таблиц переходов для асинхронных автоматов, заданных функциями переходов и выходов. При реализации любого из переходов таблицы отсутствуют опасные состязания между элементами памяти.

в) Решена задача построения множества так называемых недопустимых состояний асинхронного автомата, заданного функциями переходов и выходов. Из недопустимого состояния не гарантируется переход в устойчивое состояние без опасных

состязаний между элементами памяти.

5. Все предлагаемые алгоритмы, кроме процедур построения таблиц переходов, представлены в виде программ на языке ЛЯПАС, т.е. доведены до уровня их практического использования.

Программы образуют обширную библиотеку подпрограмм, которая находит применение при решении других задач из области диагностики неисправностей дискретных автоматов.

Развитые в диссертационной работе методы отыскания одного корня булевых уравнений могут использоваться при решении задач диагностики и контроля больших интегральных схем, для которых зачастую оказывается практически невозможным построение эквивалентных нормальных форм, а также нахождение множества всех тестов, обнаруживающих данную неисправность.

Материалы реферируемой работы нашли применение в научно-исследовательской работе "Разработка практических методов диагностики и контроля цифровых измерительных устройств на базе алгоритмического языка ЛЯПАС", выполненной для одного из предприятий в 1968-1971 гг.

Результаты работы докладывались на следующих конференциях.

1. На Первом Всесоюзном совещании по технической диагностике, Москва, 20-23 октября 1969 г.

2. На Втором Всесоюзном совещании по теории релейных устройств и конечных автоматов, Рига, 28 сентября - 1 октября 1971 г.

3. На Втором Всесоюзном семинаре по информационным методам в системах управления, измерений и контроля. Владивосток, 14-23 сентября 1972 г.

4. На XVII научно-технической конференции по радиоэлектронике. Томск, 5-7 мая 1970 г.

5. На XIX научно-технической конференции, посвященной Дню радио, Томск, 6-8 мая 1971 г.

6. На Второй научной конференции по математике и механике. Томск, 1-5 февраля 1972 г.

Основное содержание работы опубликовано в следующих статьях:

1. А.Д.Закревский, А.Ю.Калинкова. Решение системы логических уравнений. Сб. Логический язык для представления алгоритмов синтеза дискретных автоматов. "Наука", М., 1966, стр. 149-158.

2. А.Ю.Матросова. Построение проверяющих тестов для каскадных схем. Сб. Техническая диагностика. Труды I Всесоюзного совещания по технической диагностике, "Наука", М., 1972, стр. 181-184.

3. А.Ю.Матросова. О построении множества недопустимых состояний асинхронного автомата. Труды СФТИ, вып.62 Томск, 1971, стр. 11-19.

4. А.Ю.Матросова. О построении таблиц переходов для асинхронных автоматов, заданных функциями переходов и выходов. Сб.докладов 19-й научно-технической конференции, посвященной Дню радио (в печати).

5. Н.Г.Дронова, А.Ю.Матросова. Проверка булевой функции, заданной в скобочной форме, на тождественность единице. Материалы II научной конференции по математике и механике. Т.I, изд.ТГУ, 1972, стр.40-42.

6. Н.Г.Дронова, А.Ю.Матросова. Решение булевых уравнений специального вида. Сб.Информационные методы в системах управления, измерений и контроля (доклады II Всесоюзного семинара, сентябрь 1972, Владивосток), т.I, стр. 36-41.

Литература

1. C.Cherry, P.K.Vaswani. A new type of computer in prosedures. Information and Control, 1961, 4, N 2-3, 155-158.

2. Ю.Г.Григорьян. Алгоритмы решения систем логических уравнений. ЖВМ и МФ М., Изд-во АН СССР, 1962, 2, 186-189.

3. А.Д.Закревский. К решению системы логических уравнений. Принципы построения сетей и систем управления. М., Изд."Наука", 1964, 48-55.

4. А.Д.Закревский, А.Ю.Калныкова. Решение системы логических уравнений. Логический язык для представления алгоритмов синтеза релейных устройств. Изд. "Наука", М., 1966, 149-158.

5. А.Д.Закревский. К проверке тождеств в алгебре логики. Логический язык для представления алгоритмов синтеза релейных устройств. Изд. "Наука", М., 1966.

6. А.Д.Закревский. Алгоритмический язык ЯПАС и автоматизация синтеза дискретных автоматов. Изд. ТГУ, 1966.

7. А.Д.Закревский. Алгоритмы синтеза дискретных автоматов. Изд. "Наука", М., 1971.

8. J.M.Galey, R.E.Norby, J.P.Roth. Techniques for the diagnosis of switching circuits failures. Trans. IEEE Commun.and Electron., 83, Sept., 1964, 509-514.

9. J.P.Roth. Diagnosis of automata failures: a calculus and a method. IBM journal R.D., 10, N 4, 1966, 278-291.

10. А.А.Уткин. Вычисление функции различения в диагностике комбинационных схем. Автоматика и вычислительная техника, 1969, № 2, 49-53.

11. E.B.Eichelberger. Hazard detection in combinational and sequential switching circuits. IBM.J.Res. and Developm., 1965, 9, N 2.

КЗ-01631. ПОДПИСАНО К ПЕЧАТИ 31 ЯНВАРЯ 1973 г.

ЗАКАЗ 674. ТИРАЖ 120 ЭКЗ. БЕСПЛАТНО.

РОТАПРИНТ ТГУ. г. ТОМСК-29, ул. НИКИТИНА 17.