

НЕКОТОРЫЕ ВОПРОСЫ АВТОМАТИЧЕСКОГО ПОСТРОЕНИЯ СИСТЕМ ДИАГНОСТИЧЕСКИХ ТЕСТОВ ДЛЯ ЭВМ

А.В. Горбунов
(Москва)

1. Постановка задачи диагностики и вводящие замечания.

1. Всякий программный метод контроля работы вычислительной машины опирается на некоторый способ проверки её исправности. Об исправности (неисправности) машины можно судить по совпадению (несовпадению) следующих характеристик:

1.1) результатов, просчитанных в относительно одно и то же время, но разными способами (разные, но логически эквивалентные последовательности операций);

1.2) результатов, полученных одним и тем же способом (некоторая последовательность операций), но в разное время (сравнение результатов с заранее просчитанными): Строго говоря, это относится лишь к контролю так называемых устойчивых неисправностей, т.е. таких, при которых логическая функция λ_i элемента e_i , заменяясь (в результате неисправности) на функцию $\lambda'_i \neq \lambda_i$, сохраняет вид λ'_i на все последующее время работы элемента e_i .

Могут встречаться и неустойчивые неисправности, когда неисправный элемент e_i в одних случаях реализует функцию λ_i , а в других λ'_i . Однако изменением режима работы машины (например, изменением уровня срабатывания элементов и пр.) неустойчивые неисправности в большинстве случаев сводятся к устойчивым, поэтому в дальнейшем под неисправностями будут пониматься устойчивые неисправности.

В данной работе используется способ 1.2) определения неисправностей в машине (сравнение результатов с заранее просчитанными), так как он проще способа 1.1), что очень важно при ав-

томатической составлении тестов.

2. Пусть мы имеем список $\{N_1\}$ возможных неисправностей вычислительной машины и некоторую систему программ (тестов) $\{T_j\}$. Рассмотрим таблицу (см. рис. I). Строки этой таблицы соответствуют неисправности из списка $\{N_1\}$, а столбцы - тестам $\{T_j\}$. Клетки $N_1 T_j$ этой таблицы заполним по следующему правилу:

2.1) выбираем тест T_j , "проигрываем" его на исправной машине и получаем "верный" результат α_j .

2.2) Затем вносим неисправность N_1 , снова проигрываем тест T_j и получаем "пробный" результат β_j^1 .

2.3) Полагаем

$$N_1 T_j = \begin{cases} 1 & \text{при } \alpha_j \neq \beta_j^1, \\ 0 & \text{при } \alpha_j = \beta_j^1. \end{cases}$$

2.4) Восстанавливаем неисправность N_1 и переходим неисправности N_{1+1} и т.д. заполняем столбец T_j таблицы. Затем переходим к заполнению нового столбца T_{j+1} и т.д. заполняем всю таблицу.

Такая таблица называется диагностической таблицей, [1] [3].

С помощью этой таблицы задача диагностики, т.е. различения, какая из неисправностей списка $\{N_1\}$ имеет место в данный момент, может быть сформулирована так:

построить такую систему $\{T_j\}$ диагностических тестов, чтобы при построении диагностической таблицы её строки, соответствующие неисправностям, которые необходимо различать при диагностике, были бы разными.

Действительно, пусть мы имеем такую систему тестов $\{T_j\}$, тогда, "проигрывая" подряд тесты этой системы и сравнивая полученные результаты с заранее просчитанными на исправной машине, мы сможем определить какой строке таблицы соответствует данная конфигурация несовпадений результатов и, следовательно, какая неисправность списка $\{N_1\}$ имеет место в данный момент.

При этом, если в процессе ликвидации неисправности в машине заменяется весь неисправный конструктивный блок (съемный), то нет необходимости различать все неисправности списка $\{N_1\}$. Различать нужно лишь те неисправности, которые относятся к

разным съемным конструктивным блокам. В дальнейшем это обстоятельство будет учтено.

3. Заметим, что для построения диагностической таблицы требуется легко вносить и восстанавливать неисправности в машине, а также сравнивать результаты проигрывания тестов. Выполнение этих условий требует использования моделирования той машины M , для которой желательно получить тесты. В дальнейшем мы будем предполагать, что у нас имеется запрограммированная модель μ реальной машины M , такая, что:

3.1) все операции, выполняемые на машине M логически также выполняются на её модели μ .

3.2) Каждому элементу E_1 логики или памяти машины M сопоставлен некоторый элемент e_1 её модели μ .

3.3). Каждому преобразованию элементов $\{E_1\}$ машины M (например, при возникновении неисправности) сопоставлено преобразование соответствующих элементов $\{e_1\}$ модели μ (имитация неисправности), причем такое, что в результате одновременного появления таких изменений в машине M и модели μ получающиеся при этом новая машина M' и новая модель μ' снова удовлетворяют условиям 3.1) - 3.3).

Все последующие рассуждения относятся к описанной модели и её неисправностям. Для реальной машины M они будут справедливы к той степени, в какой используемая модель μ будет удовлетворять условиям 3.1) - 3.3).

Один из подходов к построению моделей вычислительных устройств, удовлетворяющих условиям 3.1)-3.3) и позволяющих автоматизировать процесс внесения и восстановления неисправностей, изложен в работе [2].

II. Об одном алгоритме отбора тестов на основе последовательного построения диагностической таблицы с малой избыточностью.

Итак, весь вопрос в том, как построить систему тестов $\{T_i\}$, о которой говорилось в формулировке задачи диагностики. Для последовательного построения этой системы, можно воспользоваться следующей схемой:

1) построение пробных тестов

2) проверка их на удовлетворение критерию отбора

- нет; отбросить неподходящий тест и вернуться к 1)

- да; перейти к 3)

3) добавление отобранных "удачных" тестов к ранее найденным.

Рассмотрим подробнее элементы этой схемы.

Построение пробных тестов может быть осуществлено с помощью случайных кодов. Поскольку система команд машины обычно содержит небольшое количество различных команд, то можно составить из них всевозможные комбинации, а при построении пробных тестов лишь менять случайные коды, на которых работают эти последовательности команд (тесты).

Рассмотрим принцип отбора пробных тестов. Пусть нам удалось выбрать первый тест T_1 . Заполним соответствующий ему первый столбец диагностической таблицы. Тогда некоторые элементы столбца получают значение "0", а другие - "1". Весь список неисправностей $\{N_1\}$ будет разбит при этом на две группы.

Приступая к отбору второго теста T_2 , будем требовать, чтобы он задавал новое разбиение списка $\{N_1\}$, отличное от разбиения заданного тестом T_1 . Пусть нам удалось выбрать такой тест T_2 . Тогда список $\{N_1\}$ будет разбит системой тестов (или столбцов) (T_1, T_2) не менее чем на три группы, и не более чем на четыре группы;

1) $T_1 = 1$ $T_2 = 1$ 3) $T_1 = 0$ $T_2 = 1$

2) $T_1 = 1$ $T_2 = 0$ 4) $T_1 = 0$ $T_2 = 0$

Введем определение;

Назовем подробностью разбиения списка $\{N_1\}$ с помощью (K) тестов (T_1, T_2, \dots, T_K) , число (n_K) групп, на которое эти (K) тестов разбивают список $\{N_1\}$.

Тогда, построив (K) тестов (T_1, T_2, \dots, T_K) и достигнув некоторой подробности n_K разбиения списка $\{N_1\}$, мы будем искать тест T_{K+1} такой, чтобы подробность n_{K+1} разбиения с помощью "дополненной" системы из $(K+1)$ тестов $(T_1, T_2, \dots, T_K, T_{K+1})$ была бы больше, т.е. удовлетворяла бы условию:

$$n_{K+1} \geq 1 + n_K.$$

При этом повышение подробности разбиения списка $\{N_i\}$ должно происходить за счет разделения неисправностей, относящихся к различным съемным конструктивным блокам B_k .

Алгоритм, реализующий сформулированный принцип отбора, работает с двумя таблицами: с отборочной таблицей и с основной диагностической таблицей (см. рис. I). Отборочная таблица равна по высоте основной таблице и состоит из столбцов, соответствующих пробным тестам τ_j . Основная диагностическая таблица служит для хранения столбцов тех тестов, которые успешно прошли отбор и были поэтому добавлены к строящейся системе тестов $\{T_j\}$.

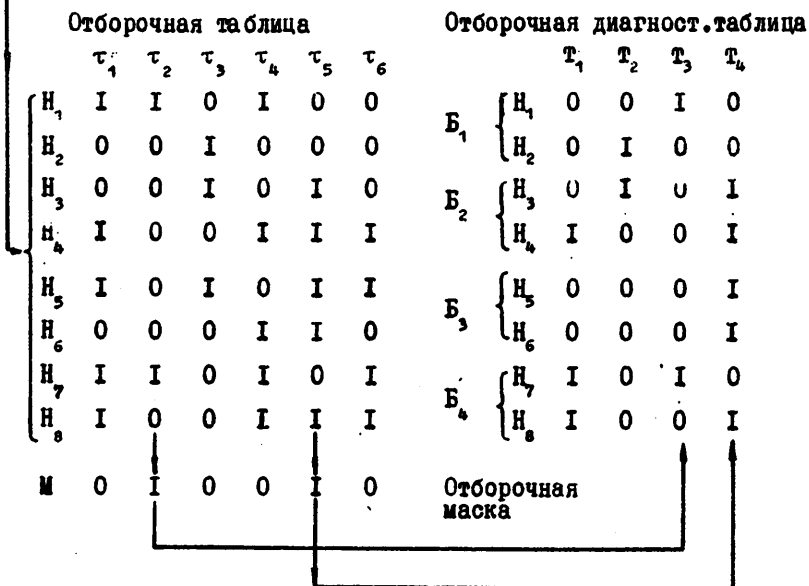
Алгоритм заключается в перенесении этих "удачных" (т.е. удовлетворяющих критерию отбора) столбцов из отборочной таблицы в основную и в добавлении "удачных" тестов к системе $\{T_j\}$. Отбор "удачных" столбцов и тестов производится на основе анализа специальной "отборочной маски". Отборочная маска представляет собой строку, равную по длине строке отборочной таблицы, но не связанную ни с какой неисправностью. Проследим процесс её заполнения.

В начале отборочная маска пуста, т.е. состоит из всех нулей. Сравним две строки N_k и N_i основной таблицы, относящиеся к разным съемным блокам. Если они различаются хоть одним элементом, то выберем две другие такие строки (например, N_k и N_{i+1}). Если мы не найдем двух одинаковых строк, относящихся к разным съемным блокам, то это будет означать, что диагностическая таблица полностью построена.

Итак, если две строки N_k и N_i из разных блоков основной таблицы полностью совпадают, то переходим к сравнению таких же строк N_k и N_i , но уже в отборочной таблице. При этом будем сравнивать их в пределах тех разрядов, отборочной маски, в которых к этому моменту уже стоят "1". В частном случае отборочная маска может состоять из одних нулей и тогда любые две строки отборочной таблицы считаются неразличимыми в данных пределах.

Если в пределах отборочной маски строки N_k и N_i отборочной таблицы неразличимы, то мы пытаемся найти вне этих пределов такой столбец τ_j отборочной таблицы, чтобы элементы $N_k \tau_j$ и $N_i \tau_j$ были бы различны ($N_k \tau_j \neq N_i \tau_j$). Если такой столбец τ_j найдется, то отборочная маска "расширяется" установкой "1" на j -ом месте. Если такого столбца найти не

Пробные тесты $\{\tau_j\}$	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6
"Верные" результаты $\{\alpha_j\}$	α_1	α_2	α_3	α_4	α_5	α_6
"Пробные" результаты $\{\beta_j^i\}$	β_1^1	β_2^1	β_3^1	β_4^1	β_5^1	β_6^1
Строка H_1 отбор.таблицы	1	0	1	0	1	1



Правило заполнения строки отборочной таблицы

$$H_1 \tau_j = \begin{cases} 0 & \text{при } \alpha_j = \beta_j^1 \\ 1 & \text{при } \alpha_j \neq \beta_j^1 \end{cases}$$

Рис. I. Схема информационных связей при построении основной диагностической таблицы.

удается (это означает, что строки H_k и H_l отборочной таблицы совпадают), то возвращаемся к основной таблице и повторяем процедуру для двух других её одинаковых строк (например H_k и H_{l+1}) из разных блоков и т.д.

Когда мы повторим эту процедуру для каждой пары одинаковых строк из разных блоков основной таблицы, то в некоторых клетках отборочной маски появятся единицы. Они укажут те столбцы отборочной таблицы, добавление которых к основной таблице позволит различать некоторые её строки, которые до этого были неразличимы (одинаковы).

Рассмотрим пример (см. рис. I). Пусть основная таблица содержит уже 2 столбца (T_1, T_2). При этом строки $(H_1, H_5, H_6), (H_2, H_3), (H_4, H_7, H_8)$ неразличимы внутри своих групп. Следуя алгоритму,

берем строку основной таблицы, соответствующую неисправности H_1 из съемного блока Б1, и сравниваем со всеми остальными строками (из других блоков). Обнаруживаем, что строка $H_5 = H_1$ и переходим к сравнению строк H_5 и H_1 отборочной таблицы в пределах отборочной маски. Так как отборочная маска пуста, то ищем такой столбец τ_1 отборочной таблицы, чтобы $H_5\tau_1 \neq H_1\tau_1$. Такой столбец существует - это τ_2 . Действительно (см. рис. I)

$H_5\tau_2 \neq H_1\tau_2$. Отмечаем найденный "удачный" столбец единичкой в клетке $M\tau_2$ отборочной маски. Далее обнаруживаем, что строка $H_6 = H_1$, однако строки H_6 и H_1 отборочной таблицы различаются в пределах отборочной маски ($H_6\tau_2 \neq H_1\tau_2$). Это

означает, что отмеченный ранее "удачный" столбец τ_2 отборочной таблицы позволит разделить не только пару $H_1 - H_5$, но и $H_1 - H_6$. Аналогично, замечая $H_2 = H_3$, так как строки H_2 и H_3 неразличимы в пределах отборочной маски ($H_2\tau_2 = H_3\tau_2$),

ищем другой столбец. Этим столбцом оказывается τ_4 . Отмечаем его единичкой в клетке $M\tau_4$ отборочной маски. Далее $H_4 = H_7$, но ($H_4\tau_2 \neq H_7\tau_2$) и, кроме того ($H_4\tau_4 \neq H_7\tau_4$).

И, наконец, $H_4 = H_8$, однако ни среди "удачных" столбцов, ни среди других столбцов отборочной таблицы нет такого, который позволил бы различить эти две неисправности. После окончания проверки мы переносим "удачные столбцы" в основную таблицу и даем "удачным" тестам τ_2 и τ_4 новые обозначения T3 и T4. Затем мы можем построить новую серию пробных тестов и соответ-

ствующую им отборочную таблицу и снова повторить процесс отбора.

Таким образом, с каждым шагом система $\{T_j\}$ диагностических тестов пополняется новыми тестами, а по основной диагностической таблице мы можем каждый раз судить, какие из пробных тестов помогут увеличить подробность разбиения списка неисправностей $\{N_i\}$. Причем строящаяся система $\{T_j\}$ и основная диагностическая таблица обладают сравнительно небольшой избыточностью. Добавляются лишь те столбцы и тесты, которые дают возможность разделить хотя бы две новые неисправности, которые до этого не различались. Что же касается той избыточности, которая остается в таблице и тестах, то она существенно связана с тем, что построение системы тестов (и таблицы) ведется последовательно. В связи с этим интересно сравнить данный алгоритм с алгоритмом Чанга.

Заметим, что алгоритм Чанга [3] точнее данного, так как он ликвидирует ту избыточность системы тестов (или, что то же, диагностической таблицы), которая допускается нашим алгоритмом. Однако эта точность получается за счет того, что алгоритм Чанга работает с таблицами, построение которых полностью закончено. Основным условием, обеспечивающим его эффективность, является возможность рассматривать и переставлять сразу все столбцы таблицы.

В нашем случае это условие не выполняется, некоторых столбцов просто нет в таблице, так как её построение еще не закончено. При этом, если не трогать столбцы уже перенесенные в основную диагностическую таблицу, а манипулировать (по Чангу) с "удачными" столбцами, выделяемыми из отборочной таблицы, то, это, как нетрудно убедиться, не дает никакого сокращения избыточности. Если не позволить на каждом шаге пересматривать по Чангу обе таблицы (основную и отборочную), то это во столько раз удлинит каждый шаг, сколько столбцов будет в этот момент в обеих таблицах. Это существенно удлинит весь процесс построения таблицы, хотя заранее трудно сказать, как это скажется на эффективности решения задачи диагностики в целом.

4. Особенности построения системы диагностических тестов с помощью диагностической таблицы, при наличии ограничений на объем памяти

При построении системы диагностических тестов с помощью диагностической таблицы, следует иметь в виду, что диагностическая таблица занимает много места в памяти и при её последовательном построении может возникнуть ситуация, когда памяти уже не хватает, а неисправности еще не все разделились.

При этом, как правило, построенная таблица обладает избыточностью, т.е. информация, позволяющая с помощью системы диагностических тестов отличать одну неисправность от другой, записывается в таблице некомпактно. Это позволяет применять различные способы оптимизации (или уплотнения) диагностических таблиц. Некоторые из этих методов (например, метод Чанга [8]) предназначены для сокращения уже построенных таблиц и поэтому избавляют лишь от части избыточности таблицы, связанной с наличием в ней некоторого числа эквивалентных столбцов или групп столбцов.

Предлагаемый здесь способ предназначен для последовательного построения больших диагностических таблиц в условиях ограниченной оперативной памяти и поэтому направлен на максимально возможное уплотнение той информации, которая должна постоянно храниться в оперативной памяти.

Рассмотрим процедуру "уплотнения" диагностической таблицы за счет "перекачки" части информации из структуры диагностической таблицы в структуру диагностических тестов, которая позволяет решить эту задачу.

Пусть мы имеем диагностическую таблицу, в которой не все неисправности разделились, т.е. в таблице имеются одинаковые строки (см. табл. D_0 на рис.2). Заменяя, где это нужно, столбцы T_j на их логические отрицания \bar{T}_j добьемся, чтобы в первой строке H_1 стояли все единицы (см. таблицу $D_{1,2}$ на рис.2). Тогда и все совпадающие с H_1 строки (например H_2) будут состоять из одних единиц, а во всех других строках найдется хотя бы один ноль. Логически перемножая между собой некоторые столбцы полученной таблицы, получим новый столбец T_1^* , у которого единицы стоят в тех строках, которые были вначале одинаковы с

$$D_0$$

	T_1	T_2	T_3
H_1	I	0	0
H_2	I	0	0
H_3	0	0	0
H_4	0	I	0
H_5	0	0	0
H_6	I	I	I
H_7	0	I	0
H_8	0	0	0

$$D_{1,2,3}$$

	\bar{T}_1	\bar{T}_2	\bar{T}_3
H_1	I	I	I
H_2	I	I	I
H_3	0	I	I
H_4	0	0	I
H_5	0	I	I
H_6	I	0	0
H_7	0	0	I
H_8	0	I	I

$$D_{4,7}$$

	\bar{T}_1	T_2	\bar{T}_3
H_1	0	0	I
H_2	0	0	I
H_3	I	0	I
H_4	I	I	I
H_5	I	0	I
H_6	0	I	0
H_7	I	I	I
H_8	I	0	I

Группы неразличимых неоправностей (строк) таблицы D_0 .

1) H_1, H_2

2) H_3, H_5, H_8

3) H_4, H_7

4) H_6

$$D_{3,5,8}$$

	\bar{T}_1	\bar{T}_2	\bar{T}_3
H_1	0	I	I
H_2	0	I	I
H_3	I	I	I
H_4	I	0	I
H_5	I	I	I
H_6	0	0	0
H_7	I	0	I
H_8	I	I	I

$$D_6 (= D_0)$$

	T_1	T_2	T_3
H_1	I	0	0
H_2	I	0	0
H_3	0	0	0
H_4	0	I	0
H_5	0	0	0
H_6	I	I	I
H_7	0	I	0
H_8	0	0	0

$$D^*$$

	T_1^*	T_2^*	T_3^*	T_4^*
H_1	I	0	0	0
H_2	I	0	0	0
H_3	0	I	0	0
H_4	0	0	I	0
H_5	0	I	0	0
H_6	0	0	0	I
H_7	0	0	I	0
H_8	0	I	0	0

Формулы связи исходных столбцов (тестов) $\{T_j\}$ со сложными столбцами (тестами) $\{T_j^*\}$

$$T_1^* = T_1 \cdot \bar{T}_3 \quad T_3^* = \bar{T}_1 \cdot T_2$$

$$T_2^* = \bar{T}_1 \cdot \bar{T}_2 \quad T_4^* = T_3$$

Рис. 2

H_1 , а во всех остальных стоят нули (см. табл. D^* рис. 2). Этот столбец T_1^* будет соответствовать такому тесту, который реагирует лишь на неисправности группы "неразличимых" неисправностей, к которой принадлежит неисправность H_1 (т.е. H_1, H_2).

Возьмем теперь (в таблице D_0) другую строку (H_3), несовпадающую с H_1 , и снова, заменяя, где нужно, столбцы T_j на \bar{T}_j , добьемся, чтобы эта строка (и все совпадающие с ней строки таблицы) состояли из одних единиц (таблица $D_{3,5,8}$ на рис.2). Логическим перемножением некоторых столбцов полученной таблицы построим новый столбец T_2^* , который содержит единицы в строках неисправностей, неразличимых с H_3 , и т.д.

Повторим процедуру для всех групп неразличимых неисправностей, на которые исходная диагностическая таблица разбила список неисправностей $\{H_i\}$. Получившиеся в результате этого столбцы $\{T_j^*\}$ содержат единички в клетках строк, соответствующих неразличимым неисправностям, входящим в данную группу. Эти столбцы будут связаны логическими формулами со столбцами исходной диагностической таблицы (см.рис. 2).

Если теперь по этим формулам скомбинировать из исходных тестов $\{T_j\}$ новую систему тестов $\{T_j^*\}$, то каждый из этих тестов будет реагировать лишь на неисправности своей группы (они отмечены единицами в соответствующем столбце T_j^*).

Чтобы прояснить значение и целесообразность такого комбинирования тестов, проанализируем роль диагностической таблицы при последовательном построении системы диагностических тестов.

Диагностическая таблица хранит при построении системы тестов информацию двух типов:

1) информацию, позволяющую при "проигрывании" уже построенной системы диагностических тестов по конфигурации несовпадений результатов с заранее просчитанными судить о том, к какой группе неразличимых неисправностей относится данная неисправность.

2) информацию, позволяющую с помощью алгоритма отбора выбирать "удачные" тесты из серии пробных тестов и таким образом продолжить построение системы диагностических тестов.

Из анализа алгоритма отбора "удачных" тестов (см. часть II этой статьи) ясно, что роль информации второго типа состоит

только в том, чтобы отличать строки одной группы неразличимых неисправностей от строк другой группы. Но такое отличие можно обеспечить, записав в строках диагностической таблицы двоичные коды порядковых номеров тех групп неразличимых неисправностей, к которым относятся эти строки. Однако при этом мы "затерли" бы информацию первого типа, хранящуюся в этих строках в виде длинных кодов из нулей и единиц. Если не удастся заранее переместить информацию первого типа из таблицы в структуру диагностических тестов, то информацию второго типа можно будет записать в таблице более компактно и таким образом освободить место в памяти. Действительно, если данная таблица позволяет различить n_1^* групп неразличимых неисправностей, то двоичный код максимального номера группы (это будет n_1^*) потребует для своей записи $(\log_2 n_1^*)^+$ двоичных разрядов, где через $(x)^+$ обозначено дополнение числа x до ближайшего целого. Точнее

$$(x)^+ = \begin{cases} x, & \text{если } x - \text{целое;} \\ \text{целая часть } (x+1), & \text{если } x - \text{дробное.} \end{cases}$$

Следовательно, такое же число столбцов потребуется для хранения этой информации в диагностической таблице. При этом диагностическая таблица сильно уплотнится, так как, например, $(\log_2 1000_{10})^+ = 10_{10}$.

В какой же форме может быть произведено перемещение информации из структуры диагностической таблицы в структуру тестов?

Из приведенного выше определения информации первого типа следует, что после проведения "перекачки" система диагностических тестов должна взять на себя функции информации первого типа, т.е. она должна при "проигрывании" не просто выдавать конфигурацию несовпадений результатов с заранее просчитанными на исправной машине, а указывать группу неразличимых неисправностей, к которой относится имеющаяся в машине неисправность (например, печатать или подавать на пульт номер этой группы неисправностей).

После этих рассуждений становится понятным наше стремление скомбинировать из исходных тестов $\{T_j^*\}$ систему $\{T_j^*\}_1$ сложных (комбинированных) тестов, каждый из которых реагирует лишь на неисправности, входящие в некоторую группу неразличимых

ных неисправностей исходной диагностической таблицы и при этом указывает номер этой группы.

Рассмотрим способ, которым это может быть сделано.

Предварительно преобразуем каждый тест T_j исходной системы тестов к некоторому стандартному виду (см. рис. 3 а), б)).

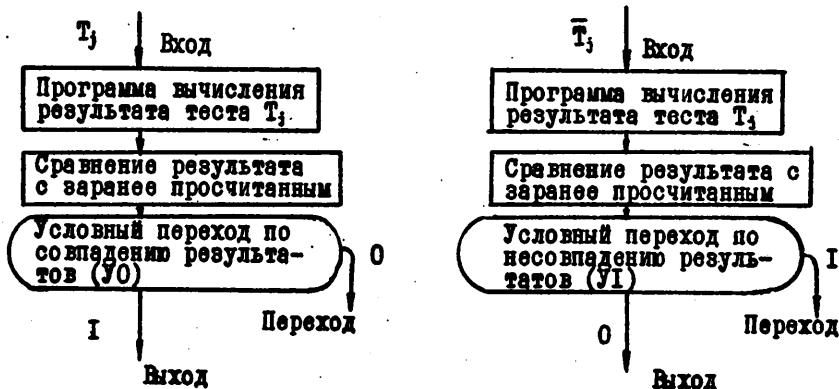


Рис. 3. Стандартный вид:

а) "прямого" теста T_j ,

б) "обращенного" теста \bar{T}_j .

рис. а) следует, что когда тест реагирует на неисправность (несовпадение результата "проигрывания" теста с заранее просчитанным), управление передается на следующую команду (на рис. отмечено стрелкой "выход"). В случае, когда тест не реагирует на неисправность (совпадение результатов), происходит условный переход по совпадению результатов (на рис.а) отмечено стрелкой "переход"). В дальнейшем будет уточнено, куда в этом случае передается управление. А сейчас отметим, что такой тест T_j будет соответствовать столбцу T_j (т.е. будет реагировать лишь на те неисправности, которые отмечены единицами в столбце T_j исходной диагностической таблицы). Такие тесты мы будем называть "прямыми" в отличие от "обращенных" тестов \bar{T}_j , которые соответствуют столбцам \bar{T}_j , получающимся при логическом отрицании столбцов T_j . Легко видеть, что для получения "обращенного" теста \bar{T}_j из "прямого" теста T_j (в стандартной форме) достаточно заменить условный переход по совпадению резуль-

татов (см. рис. 3 б). В более схематичном виде "прямой" тест и обращенный тест представлены на рис. 4 а,б. Здесь стрелка

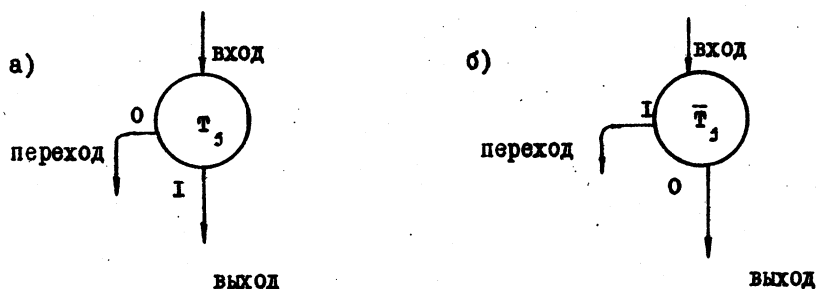


Рис. 4 Схематическое изображение

а) "прямого" теста T_j

б) "обращенного" теста \bar{T}_j

"выход" обозначает передачу управления на следующую команду:

"1" - при несовпадении результатов,

"0" - при совпадении результатов.

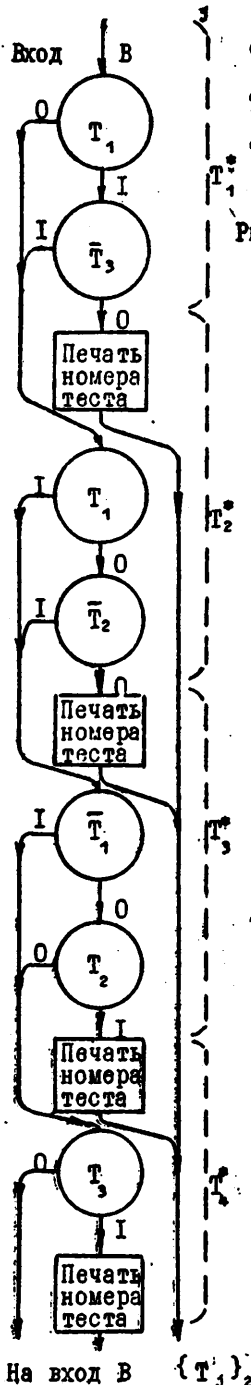
Стрелка "переход" обозначает условный переход:

"1" - по несовпадению результатов,

"0" - по совпадению результатов.

На рис. 5 изображен пример комбинирования системы $\{T_j^*\}_1$ сложных тестов из тестов исходной системы $\{T_j\}$, представленных в стандартной форме, и их логически "обращенных" тестов

$\{\bar{T}_j\}$ (по формулам рис.2). Стрелки "переход" с индексами "0" и "1" в сложном тесте T_j^* все направлены по входу в очередной сложный тест T_{j+1}^* . Это связано с тем, что каждый сложный тест T_j^* реагирует (т.е. выдает номер группы неисправностей) лишь на определенную конфигурацию реакций, входящих в него исходных тестов T_j и обращенных тестов \bar{T}_j . Эти конфигурации и есть носители информации первого типа, перенесенной из диагностической таблицы в тесты. Выход из сложного теста T_j^* по стрелке "переход" означает, что данный сложный



Формулы комбинирования (см. также рис. 2)

$$T_1^* = T_1 \cdot \bar{T}_2, \quad T_3^* = \bar{T}_1 \cdot T_2$$

$$T_2^* = \bar{T}_1 \cdot \bar{T}_2, \quad T_4^* = T_3$$

Рис. 5. Пример построения системы сложных (комбинированных) тестов $\{T_j^*\}$ из исходных тестов $\{T_j\}$ и их логически "обращенных" тестов $\{\bar{T}_j\}$.

тест уже не сможет отреагировать на получающуюся конфигурацию реакций (несовпадения результатов), входящих в него тестов. Поэтому целесообразно сразу же переходить к проверке реакций следующего сложного теста T_{j+1}^* и т.д. до тех пор, пока

какой-нибудь из сложных тестов не отреагирует на получающуюся конфигурацию реакций, входящих в него исходных тестов. При этом будет выдан (на печать или на пульс) номер этого теста, т.е. номер группы различных неисправностей, в которую входит имеющаяся неисправность.

Таким образом, комбинированная из исходных тестов T_j и их логически "обращенных" тестов \bar{T}_j , система сложных тестов $\{T_j^*\}$, способна хранить всю информацию первого типа, которая была накоплена в диагностической таблице. Причем, так как эта информация не нужна для продолжения построения системы диагностических тестов, то её не нужно хранить в оперативной памяти и, следовательно, объем памяти, требуемый для хранения системы сложных тестов $\{T_j^*\}$, не так существенен.

Информацию второго типа, необходимую для продолжения построения системы диагностических тестов, запишем в следующем ви-

де: в каждой строке H_1 таблицы запишем двоичный код номера j того сложного теста T_j^* , который реагирует лишь на неисправности данной группы неразличимых неисправностей исходной таблицы (в которую входит сама неисправность H_1). Это, как было показано, займет лишь $(\log_2 n_1^*)^+$ столбцов основной диагностической таблицы. Освободившуюся при этом память используем для достраивания основной диагностической таблицы с учетом ранее накопленной информации второго типа, записанной в сжатом виде в первых $(\log_2 n_1^*)^+$ столбцах.

Если список неисправностей $\{H_1\}$ велик, то рано или поздно мы снова заполним всю память, отведенную под основную диагностическую таблицу, а многие неисправности при этом еще останутся неразделенными. Тогда, если вновь построенная часть основной диагностической таблицы (без учета "старых" $(\log_2 n_1^*)^+$ столбцов) задает разбиение списка $\{H_1\}$ неисправностей на n_2^* групп, то можно снова переместить новую информацию первого типа в структуру новой системы $\{T_j^*\}_2$ сложных тестов, а новую информацию второго типа уплотнить и записать в $(\log_2 n_1^*)^+$ столбцах, следующих сразу за "старыми" $(\log_2 n_1^*)^+$ столбцами основной диагностической таблицы и т.д.

Подобную процедуру можно повторять до тех пор, пока не закончится разделение всех неисправностей (которые следует разделять), или, по крайней мере, до тех пор, пока

$$\sum_k (\log_2 n_k^*)^+ < N_{ст.}$$

где n_k^* - подробность разбиения списка $\{H_1\}$ неисправностей с помощью "новой" k -ой части основной диагностической таблицы, построенной между $(k-1)$ -м и k -м повторениями процедуры перемещения и уплотнения;

N - число столбцов основной диагностической таблицы, которое уместается целиком в памяти, отведенной под эту таблицу. После последнего повторения процедуры перемещения и уплотнения мы получим таблицу, которую уместно назвать обобщенной диагностической таблицей (рис.6). Каждая строка этой таблицы будет соответствовать некоторой неисправности из списка $\{H_1\}$, а каждый столбец будет соответствовать некоторой системе $\{T_j^*\}_k$ сложных тестов. Каждый такой столбец удобно называть обобщенным столбцом, а каждую систему $\{T_j^*\}_k$ - обобщенным тестом.

Элементами обобщенного столбца $\{T_j^*\}_k$ будут номера групп, на которые обобщенный тест $\{T_j^*\}_k$ разбивает список $\{H_1\}$. Та-

	$\{T_j^*\}_1$	$\{T_j^*\}_2$	$\{T_j^*\}_3$
H_1	01	01	01
H_2	01	02	02
H_3	02	03	03
H_4	03	04	04
H_5	02	05	05
H_6	04	06	06
H_7	03	04	07
H_8	02	04	03

Рис. 6

Пример обобщенной диагностической таблицы.

ким образом, каждая строка H_i обобщенной диагностической таблицы представляет собой последовательность номеров групп, в которые входит неисправность H_i при различных разбиениях списка $\{H_i\}$, задаваемых обобщенными тестами $\{\{T_j^*\}_k\}$.

Эти разбиения не являются независимыми. Каждое последующее (в порядке возрастания индекса k) уточняет предыдущее. При этом, если обозначить

n_1 — подробность разбиения списка $\{H_i\}$ с помощью системы из 1 обобщенных тестов $(\{T_j^*\}_1, \{T_j^*\}_2, \dots, \{T_j^*\}_1)$, то подробность n_{1+1} разбиения

списка $\{H_i\}$ с помощью $(1 + 1)$ обобщенных тестов

$$(\{T_j^*\}_1, \{T_j^*\}_2, \dots, \{T_j^*\}_1, \{T_j^*\}_{1+1})$$

удовлетворяет условию

$$n_{1+1} \geq n_1 + P_{1+1},$$

где P_{1+1} — число столбцов основной диагностической таблицы, уплотнением и перемещением которых были получены обобщенный столбец и обобщенный тест $\{T_j^*\}_{1+1}$.

С помощью этой оценки нетрудно рассчитать наименьшее гарантированное число неисправностей (или групп неисправностей), на которые можно разбить список $\{H_i\}$, применяя изложенный метод, если число N столбцов, уместающихся в памяти, отведенной под основную диагностическую таблицу, задано. Или, наоборот, по заданному числу неисправностей (или групп неисправностей), которые нужно разделить, можно указать то количество столбцов, которое должно уместаться в памяти, отведенной под основную диагностическую таблицу, с тем, чтобы успешность такого разделения неисправностей можно было бы гарантировать. Разумеется, эти гарантии сохраняются лишь при условии, что заполнение основной диагностической таблицы обеспечивается (например, по способу, изложенному в части II этой статьи).

Таблица зависимости числа N столбцов основной таблицы от требуемой подробности n разбиения списка $\{H_i\}$ приведена на рис. 7.

Вернемся к рассмотрению обобщенной диагностической таблицы. Её использование при диагностике неисправностей можно предста-

$N_{\text{ст.}}$	50	100	150	200	250
n	270	800	1700	2780	4150

Рис. 7

Таблица зависимости числа $N_{\text{ст.}}$ столбцов основной диагностической таблицы от подробности n разбиения описки.

вить следующим образом: обобщенные тесты $\{T_k^*\}_k$ "проигрываются" подряд на проверяемой машине и каждый из них выдает номер группы, к которой относится имеющаяся в машине неисправность. Рассматривая эти номера в порядке возрастания индекса (k) обобщенных тестов $\{T_k^*\}_k$, выдавших эти номера, мы получим строку цифр.

Обращаясь теперь к обобщенной диагностической таблице, найдем в ней такую же строку, и та неисправность или группа неисправностей, которым соответствует эта строка таблицы, даст нам искомый результат диагностики.

В заключение автор считает приятным долгом поблагодарить организатора семинара за приглашение, а также участников семинара Папернова А.А. и Панчикова В.С. за ценные замечания, которые оказали влияние на окончательный текст работы.

ЛИТЕРАТУРА

1. "Диагностика неисправностей вычислительных машин". (Сборник), "Наука", 1965.
2. Кузьмичев Д.А. "Методы диагностики вычислительных машин" (кандидатская диссертация, ИТМ и ВТ Москва).
3. Chang Herbert I. An algorithm for selecting an optimum set of diagnostic tests. - IEEE Trans., 1965, vol. EC-14, N 5, October.