**Метод обработки тестовых артефактов для тестирования многофункциональных индикаторов авиационного бортового оборудования с использованием регулярных выражений при синтаксическом анализе.**

Бортовое радиоэлектронное оборудование современного боевого летательного аппарата (ЛА) выполняет огромное количество многочисленных задач по обеспечению пилотирования, навигации, коммуникации и безопасности. Объем программного обеспечения таких систем с каждым годом увеличивается и усложняется, в следствии чего растет потребность в тщательном тестировании.

Основным источником информации для летчика боевого ЛА являются многофункциональные индикаторы (МФИ), оснащенные вычислительной системой. Проверка функционального программного обеспечения таких индикаторов производится при помощи тестового автомата – программное обеспечение, позволяющее проводить комплексное тестирование МФИ в соответствии со сценариями (определенные последовательности действий). При выполнении проверки, программа подает на вход индикатора тестовые артефакты, представленные в виде набора сигналов, в следствие которых МФИ производит отображение информации. Информация о наборах содержится в файлах специального формата в виде таблиц информационного взаимодействия.

Обработка таблиц информационного взаимодействия основывается на проверке соответствия спецификации требований к изложению, построению и оформлению. Отличительная особенность спецификации заключается в четкой постановке инструкций по построению таблиц наборов сигналов, поэтому основная идея метода обработки тестовых артефактов лежит в использовании шаблонов.

Конструкции шаблонов строятся на основе отношений – неких универсальных связей, установленных инструкциями или разработчиком. Связи являются бинарными, т.е. переходы осуществляются от одного узла к другому, а в качестве узлов подразумеваются неделимые единицы. Если существуют два различных узла *А* и *В*, и между ними присутствует универсальная бинарная связь, то такая связь является зависимостью. В совокупности, множество таких зависимостей должно покрывать все возможные связи между значениями.

*Базовый шаблон* – установленное правило, определяющее зависимость в анализируемых ячейках таблицы. Конструкция шаблона формируется из последовательности неделимых единиц, слов или знаков. Характеризуется уникальным идентификатором. Для каждого базового шаблона определяется набор альтернативных. *Альтернативный шаблон* (формула 1.1) – установленное правило, противоречащее базовому, которое позволяет определять причину нарушения зависимости. Отличительной особенностью является наличие производимой операции при обнаружении в ячейке таблицы. Кроме того, данные шаблоны обладают приоритетом.

*Приоритет* – формальное свойство, влияющее на очередность расположения конструкций в наборе. При одинаковом приоритете шаблоны располагаются исходя из алфавитной сортировки по идентификатору или усмотрению разработчика. Установка очередности альтернативных конструкций зависит от значимости причины нарушения зависимости.

*Операция* – действие, выполняемое при соответствии конструкции ячейки таблицы одному из шаблонов. Как правило, этим действием является запись ошибки в определенную структуру, с помощью которой потом формируется отчет об обработке тестового артефакта. Если отличия базового шаблона и альтернативного незначительно, то действием возможно автоматическое исправление значения.

Обработка значений таблицы в анализируемом тестовом артефакте строится на основе поиска схожей конструкции в множестве шаблонов. Пусть существует набор значений, множество   
базовых шаблонов. Тогда для каждого отдельного базового шаблона , где определено множество альтернативных шаблонов.

Все значения из набора *P* последовательно сопоставляются с выделенными отдельными базовыми шаблонами. При соответствии конструкций производится переход к следующему значению таблицы. В противном случае происходит последовательный поиск альтернативного шаблона из соответствующего множества . После обнаружения совпадения с альтернативной конструкцией происходит выполнение операции, закрепленной для шаблона . В случае, если совпадений с альтернативным шаблоном из множества не обнаружено, выдается сообщение о присутствии неизвестной ошибки.

Таких совпадений может быть несколько, поэтому сопоставление происходит до тех пор, пока не будут проверены все шаблоны. Для окончания работы всего алгоритма необходимо обработать последовательно все значения таблицы.

Базовые и альтернативные шаблоны составляются на основе регулярных выражений, поэтому рассмотрим процесс составления шаблонов на примере одного значения. В таблицах информационного взаимодействия для каждого набора сигналов имеется определенный параметр – используемые разряды. *Используемые разряды* – набор битов, которые требуются для упаковки параметра в слово. Каждое слово состоит из 32 битов. При составлении альтернативных шаблонов для данного параметра, использовано 4 уровня приоритета, где 1-ый считается наивысшим. Согласно спецификации требований к изложению, построению и оформлению, данное значение синтаксически должно выглядеть следующим образом:

Также, в спецификации указано, что если данный параметр упаковывается в двух словах, то в одной ячейке должно содержаться два промежутка, по промежутку для каждого слова соответственно. Исходя из требований, составим базовый шаблон для параметра «используемые разряды».

Составив базовый шаблон (формула 1.3), необходимо аналитическим путем определить набор ошибок, которые может допустить человек, составляющий таблицы информационного взаимодействия при записи параметра используемых разрядов. На основе них будет составляться набор альтернативных шаблонов для поиска нарушения зависимости. Самыми распространенными ошибками будем считать: отсутствие значения в ячейке и использование недопустимых символов. Альтернативные шаблоны представлены в формулах 1.4 и 1.5 соответственно. Данные ошибки будут обладать наивысшим приоритетом (приоритет 1-го уровня), и при их обнаружении необходимо составить запись об этом в базу замечаний.

После того, как мы убедились, что ячейка не пуста и значение не содержит недопустимых символов, стоит обратить внимание на правильное обозначение промежутков. Из формулы 1.2 видно, что промежуток должен состоять из троеточия, поэтому альтернативный шаблон, указывающий на неправильное обозначение промежутка будет выглядеть следующим образом:

Данная ошибка соответственно имеет меньший приоритет (приоритет 2-го уровня), отличие от базового шаблона незначительно, поэтому при совпадении с данным шаблоном под операцией будет использоваться автоматическое исправление ошибки.

Дальше стоит обратить внимание на количество данных промежутков. Как уже было сказано выше, согласно спецификации требований к изложению, построению и оформлению, если параметр упаковывается в двух словах, то в одной ячейке должно содержаться два промежутка, по промежутку для каждого слова. Для проверки требуется составить два альтернативных шаблона: на проверку числа разделителей между промежутками и на проверку числа самих промежутков. Согласно спецификации, разделителями промежутков считается знак «,», и в одном значении содержится в единичном экземпляре. В формуле 1.7 представлен альтернативный шаблон для проверки количества разделителей:

Аналогично формуле 1.7, составляется шаблон для проверки количества промежутков:

Единственным отличием является количество промежутков, которых может быть несколько. Данные ошибки обладают приоритетом 3-го уровня, а при их обнаружении необходимо составить запись в базу замечаний.

Самым низким приоритетом (приоритет 4-го уровня) обладают ошибки об отсутствие значений до и после промежутков. Для них составляется по несколько альтернативных шаблонов для случая с одним промежутком и несколькими. При обнаружении автоматическое исправления невозможно, поскольку значение битов неизвестно, поэтому необходимо составить замечание об ошибке и записать в соответствующую базу.

Рассмотрим вариант с одним промежутком. Чтобы проверить отсутствие значения до запятой, необходимо воспользоваться альтернативным шаблоном в формуле 1.9. Для проверки значения после запятой используется формула 1.10.

Для проверки отсутствия значений в нескольких промежутках, необходимо использовать формулы 1.11 (отсутствие в первом промежутке начального значения), 1.12 (отсутствие в первом промежутке конечного значения), 1.13 (отсутствие во втором промежутке начального значения), 1.14 (отсутствие во втором промежутке конечного значения).

Метод обработки протоколов информационного взаимодействия для тестирования многофункциональных индикаторов авиационного бортового оборудования с использованием регулярных выражений применяется только к синтаксическому анализу, так как не имеет возможности проверить смысловую нагрузку набора сигналов (к примеру, перекрытие используемых разрядов или корректную упаковку данных). В следствии этого использование этого алгоритма подразумевается лишь в совокупности с методами проверки логической взаимосвязи.