

между программой и ее спецификацией, описывающей цель разработки.

Определение верификации симметрично ввиду относительного характера свойства корректности: либо программа, либо спецификация, либо оба этих объекта могут содержать ошибки, если нет соответствия между программой и спецификацией. Переходя на логическую точку зрения, можно сказать, что верификация — вычисление истинности предиката от двух аргументов: программы (на языке программирования) и спецификации (на языке спецификаций). Истинность этого предиката устанавливает свойство корректности программы, если спецификация не содержит ошибок, однако его ложность, вообще говоря, не означает наличие ошибок в программе, а требует более тонкого разбора. Перефразируя известное изречение Э. Дейкстры (о тестировании) можно сказать, что верификация способна доказать отсутствие ошибок в программе, но не всегда доказывает их присутствие.

В отличие от тестирования верификация предполагает аналитическое исследование свойств программы по ее тексту (без выполнения программы), поэтому цель верификации можно достичь только строгим математическим доказательством соответствия программы ее спецификации. Для возможности проведения таких доказательств необходимо построить соответствующую формальную систему, в которой были бы формально определены свойства корректности программы.

Учитывая специфику проявления ошибок в программах в процессе их выполнения на ЭВМ, целесообразно выделить в понятие корректности программы два свойства:

1) *частичную корректность* — удовлетворение внешним спецификациям вход-выход при условии завершения выполнения программы, т. е. достижения в процессе вычисления выхода программы;

2) *завершение* — достижение в процессе выполнения программы, начатой в допустимом (для входной спецификации) состоянии выхода программы.

Свойство завершения не является тривиальным для такого объекта, как программа, ввиду возможности циклических или рекурсивных вычислений, а также наличия частично-определенных операций. Случаи, когда свойство завершения не удовлетворяется, довольно обычны для программ с ошибками, и дают заведомо некорректные программы¹.

Каждое из двух выделенных свойств корректности программы может удовлетворяться или не удовлетворяться. Таким образом, можно выделить шесть основных задач анализа корректности:

1) *частичная корректность (при условии завершения)*;

¹ Доказательство незавершения обычно дает в качестве результата отсутствие корректности программы, независимо от спецификации выхода.