

Идея инварианта состоит в том, чтобы указать некоторые свойства процессов выполнения программы, приписываемые конкретным объектам структуры программы, например дугам управляющего графа программы, т. е. сечениям между двумя связанными (по управлению) операторами программы. Эти сечения являются точками начала или конца операторов программы. Поскольку оператор может изменять состояние памяти при выполнении программы, инвариантом сечения должно быть множество возможных состояний памяти, которые могут иметь место, когда при выполнении программы достигается данное сечение.

Для описания множества допустимых состояний в сечениях управляющего графа программы естественно использовать предикаты, истинные на данных множествах. Таким образом, мы приходим к важной идее логической характеристики поведения программы (и ее составных частей) с помощью инвариантных утверждений. Инвариантным утверждением (инвариантом) сечения s управляющего графа программы $Prgm$ называется формула логического языка спецификации $\varphi_s(x_1, \dots, x_n)$, где x_1, \dots, x_n — программные переменные, такая, что значение φ_s истинно для набора значений программных переменных, имеющего место всякий раз, когда выполнение $Prgm$ достигает данного сечения s . Инвариантность утверждения означает независимость его истинности от пути, по которому достигается данное сечение (точка).

Общность понятия инвариантного утверждения состоит в том, что оно применимо к любому сечению управляющего графа программы, в частности к началу программы (сечению, непосредственно следующему за оператором START) и к концу программы (сечению, непосредственно предшествующему оператору STOP). В сечениях начала и конца программы инвариантное утверждение задает соответственно предусловие и постусловие программы, характеризующее область исходных данных и целевую функцию программы. В промежуточных сечениях инвариантное утверждение — характеристика некоторой подцели, достигаемой в процессе вычислений по программе.

Инвариант сечения управляющего графа — это еще не законченная характеристика свойств программы или ее фрагмента. Следующим шагом является введение формализма, характеризующего свойства оператора программы с точки зрения его воздействия на состояние памяти. Таким формализмом служит тройка Хоара $\{P\}A\{Q\}$, где P, Q — инварианты входа и выхода оператора A (пред- и постусловие оператора A).

Неформальный смысл тройки Хоара состоит в следующем: если P есть истина непосредственно перед выполнением оператора A , и выполнение оператора A завершается, то Q есть истина для значений переменных, соответствующих завершению A . Формально тройка Хоара определяет предикат