

его можно представить как кортеж значений переменных программы

$$st = \langle st_1, st_2, \dots, st_m \rangle.$$

Динамический характер st проявляется в том, что в разных состояниях могут быть активизированными (доступными) различные переменные. Поэтому, вообще говоря, в множестве значений переменной st_i должно быть представлено и неопределенное значение ω_i .

Область определения функции f интерпретируется, таким образом, как многосортная область $D \subseteq D_1 \times \dots \times D_i \times \dots \times D_k$, где $st_i \in D_i$, а сама функция f есть отображение вида $State \rightarrow State$.

2. Предусловие и постусловие программы определяются с учетом дополнительного интенционального свойства — завершения вычислений, т. е. достижения в процессе выполнения программы оператора окончания программы. При этом следует иметь в виду, что для некоторых начальных состояний $st' \in State$ выполнение программы может заиклиться, либо привести к операции с неопределенным результатом. Ясно, что такие состояния st' не могут входить в область определения функции f . В результате область определения D может отличаться от $State = D_1 \times D_2 \times \dots \times D_k$, т. е. функция f , вообще говоря, есть частичная функция (неполностью определенная). Это отражает общую природу вычислимых функций как функций частично-рекурсивных.

Проведенное выше рассмотрение показывает, что логическая спецификация программы $Prgm$, характеризующая ее внешнее поведение (функционирование), может быть определена на множестве состояний памяти предикатами

$$pre-Prgm : State \rightarrow Boolean,$$

$$post-Prgm : State \times State \rightarrow Boolean$$

при условии

$$\forall st (pre-Prgm(st) \wedge fin(Prgm, st) \Rightarrow post-Prgm(st, f(st))),$$

где $st \in State$ — начальное (входное) состояние $Prgm$; $f(st) \in State$ — выходное состояние (результат выполнения) программы $Prgm$; $fin(Prgm, st)$ — предикат завершения выполнения $Prgm$.

Предикаты $pre-Prgm$ и $post-Prgm$, хотя и играют ключевую роль в спецификации программы, но, как будет видно в дальнейшем, недостаточны для аналитического исследования свойств программы. Они характеризуют лишь вход-выход программы. Для более полной логической спецификации свойств программы могут потребоваться характеристики ее поведения во «внутренних» точках структуры. Поэтому обобщим теперь намеченный подход к логической спецификации введением понятия инварианта (инвариантного утверждения).