ПРОПОЗИЦИОНАЛЬНЫЕ МЕТОДЫ ЗАДАНИЯ СЕМАНТИКИ

Пропозициональные методы задания семантики языков программирования обеспечивают более высокий уровень абстракции по сравнению с операционными определениями.

Основным множеством объектов в пропозициональных методах являются не множества состояний, как в операционных методах, а **множества формул некоторой логической системы**. Эти формулы всегда можно проинтерпретировать как некоторые утверждения относительно состояний.

Таким образом, **операторы (команды)** языка программирования могут рассматриваться в пропозициональных методах как **преобразователи формул (утверждений)**, записанных в некоторой логической системе (например, в исчислении предикатов первого порядка), либо, в более общем случае, как отношения между утверждениями.

ПРОПОЗИЦИОНАЛЬНЫЕ МЕТОДЫ ЗАДАНИЯ СЕМАНТИКИ

Наиболее известными пропозициональными методами являются:

- **Метод индуктивных утверждений Флойда для проверки правильности программ.**

- **Аксиоматический метод Хоара.**

Подробно рассмотрим аксиоматический метод Хоара, который в отличие от метода Флойда не требует предварительной трансляции текста программы в блок-схему, а позволяет работать непосредственно с текстом программы.

АКСИОМАТИЧЕСКИЙ МЕТОД ХОАРА

В аксиоматическом методе Хоара операторы языка программирования ассоциируются с отношениями между утверждениями.

Элементарным операторам сопоставляются аксиомы (или схемы аксиом) вида

$$\{P\}\ C\ \{Q\},$$

где **P** и **Q** – утверждения в некоторой логической системе, **C** – команда (программа) в языке программирования.

Эта аксиома означает, что если утверждение **P** истинно до выполнения команды **C**, то утверждение **Q** будет истинно после выполнения команды **C**.

Составным операторам сопоставляются правила вывода с одной или несколькими посылками.

АКСИОМАТИЧЕСКИЙ МЕТОД ХОАРА

Рассмотрим задание аксиоматической семантики на примере **языка L**, описанного в предыдущей лекции.

В языке **L** есть только один элементарный оператор (оператор присваивания) и его смысл задается схемой аксиом вида:

(A1) $\{P[E/x]\}\ x := E\ \{P\}$,

где **P** – утверждение в некоторой логической системе, **P[E/x]** обозначает результат подстановки выражения **E** вместо всех свободных вхождений переменной **x** в **P**.

Смысл этой аксиомы: «Если утверждение **P**, в которое вместо **x** подставлено **E**, истинно на векторе состояния, предшествующем выполнению команды **x := E**, то **P** будет истинно в состоянии, следующем за выполнением этой команды».

ПРАВИЛА ВЫВОДА

Смысл составных команд определяется правилами вывода:

$$\{P\}\ C_1\ \{Q\},\ \{Q\}\ C_2\ \{R\}$$

(R2) ———————————————

$$\{P\}\ C_1;C_2\ \{R\}$$

Это правило означает следующее: «Если истинность утверждения **P** гарантирует истинность утверждения **Q** после выполнения команды C_1, и истинность утверждения **Q** гарантирует истинность **R** после выполнения команды C_2, то истинность **P** гарантирует истинность **R** после выполнения команды $C_1;C_2$.

ПРАВИЛА ВЫВОДА

$$\{P \,\&\, B\}\ C_1\ \{Q\},\ \{P \,\&\, \neg B\}\ C_2\ \{Q\}$$

(R3)

$$\{P\}\ \underline{\text{if}}\ B\ \underline{\text{then}}\ C_1\ \underline{\text{else}}\ C_2\ \underline{\text{fi}}\ \{Q\}$$

Это правило означает следующее: «Если истинность утверждения **P & B** гарантирует истинность утверждения **Q** после выполнения команды $\mathbf{C_1}$, а истинность утверждения **P & ¬B** гарантирует истинность **Q** после выполнения команды $\mathbf{C_2}$, то истинность **P** гарантирует истинность **Q** после выполнения условного оператора».

ПРАВИЛА ВЫВОДА

$$\frac{\{P \,\&\, B\}\ C\ \{P\}}{\{P\}\ \underline{\textbf{while}}\ B\ \underline{\textbf{do}}\ C\ \underline{\textbf{od}}\ \{P \,\&\, \neg B\}}\ (R4)$$

Это правило означает следующее: «Если истинность условия **B** гарантирует сохранение истинности **P** после выполнения команды **C**, то истинность **P** сохраняется оператором цикла **while B do C od**. Кроме того, после выполнения цикла условие **B** будет ложным».

Утверждение **P** называют **«инвариант цикла»**.

ПРАВИЛО ЗАКЛЮЧЕНИЯ

$$\textbf{(R5)}\quad \frac{P \to R,\ \{R\}\ C\ \{G\},\ G \to Q}{\{P\}\ C\ \{Q\}}$$

В этом правиле используются посылки вида **A → B**, означающие, что из истинности утверждения **A** следует истинность утверждения **B**.

Данное правило позволяет пользоваться дедукцией при доказательстве свойств программ.

НЕПРОТИВОРЕЧИВОСТЬ ФОРМАЛЬНОЙ СИСТЕМЫ ХОАРА

Обозначим через **H** формальную систему Хоара.

Пусть **A** множество утверждений.

Запись $\mathbf{A} \vdash_{\mathbf{H}} \mathbf{\{P\}\ C\ \{Q\}}$ обозначает тот факт, что существуют доказательства формулы **{P} C {Q}** в формальной системе **H**, которые используют в качестве предположений (посылок) утверждения из **A**.

Формула вида **{P} C {Q}** может рассматриваться как программа с утверждениями.

При этом встает вопрос о том, что означает истинность программы с утверждениями.

Для ответа на этот вопрос необходимо ввести понятие интерпретации.

ПОНЯТИЕ ИНТЕРПРЕТАЦИИ

Зафиксируем язык ***L***, в котором записываются все утверждения относительно программы. Пусть, в частности, ***L*** есть язык первого порядка с равенством.

Пусть **I** – интерпретация ***L*** на непустой области **D**.

Под состоянием будем понимать функцию, обозначаемую через **σ**, возможно, с индексом, которая присваивает каждой переменной **x** значение из области **D**.

Определение 1. Запись $\vdash_I \mathbf{P(\sigma)}$ означает, что при интерпретации **I** утверждение **P** истинно в состоянии **σ**. Если для всех состояний **σ** имеет место $\vdash_I \mathbf{P(\sigma)}$, то говорят, что **P** истинно при интерпретации **I**, и это записывается как $\vdash_I \mathbf{P}$.

ИСТИННОСТЬ ПРОГРАММЫ

С каждой программой **C** можно ассоциировать значение $\mathbf{M_I(C)}$ при интерпретации **I**, где $\mathbf{M_I(C)}$ является частичной функцией из состояния в состояние, т.е. имеет вид $\mathbf{M_I(C): S \to S}$, где **S** – множество всех состояний.

Теперь можно определить истинность программы с утверждениями при интерпретации **I**.

Определение 2. Программа с утверждениями **{P} C {Q}** истинна при интерпретации **I**, если для всех состояний $\boldsymbol{\sigma_1}$, $\boldsymbol{\sigma_2}$ из $\mathbf{\vdash_I P(\sigma_1)}$ и $\mathbf{M_I(C)\sigma_1 = \sigma_2}$ следует, что $\mathbf{\vdash_I Q(\sigma_2)}$.

Это определение является корректной формализацией неформального понятия формулы **{P} C {Q}**.

Определение 3. Программа с утверждениями **{P} C {Q}** называется правильной, если она истинна при всех интерпретациях **I**.

ТЕОРЕМА О НЕПРОТИВОРЕЧИВОСТИ

Определение 4. Правило вывода называется непротиворечивым, если для всех интерпретаций **I** оно сохраняет истинность программ с утверждениями.

Легко доказать, что все аксиомы системы **H** истинны и правила вывода непротиворечивы, т.е. сохраняют истинность.

Из этого факта следует **теорема о непротиворечивости** формальной системы **H**: Для всякой интерпретации **I**, множества утверждений **A** и программы с утверждениями **φ** имеет место следующее: если все утверждения из **A** истинны при интерпретации **I** и $\vdash_H \varphi$, то **φ** – истинна при интерпретации **I**.

Другими словами, формальная система непротиворечива, если из $A_I \vdash_H \varphi$ следует, что $\vdash_H \varphi$, где A_I обозначает множество всех утверждений, истинных при интерпретации **I**.

ПОНЯТИЕ ПОЛНОТЫ ФОРМАЛЬНОЙ СИСТЕМЫ

Симметричным понятием для непротиворечивости является понятие полноты формальной системы.

Говорят, что формальная система **H** полна, если из $\vdash_H \varphi$ следует, что $A_I \vdash_H \varphi$.

Заметим, что формальная система **H**, определенная выше, является полной.