

МЕТРИКИ, ОСНОВАННЫЕ НА
ЛЕКСИЧЕСКОМ АНАЛИЗЕ
ПРОГРАММ. МЕТРИКИ ХОЛСТЕРА.

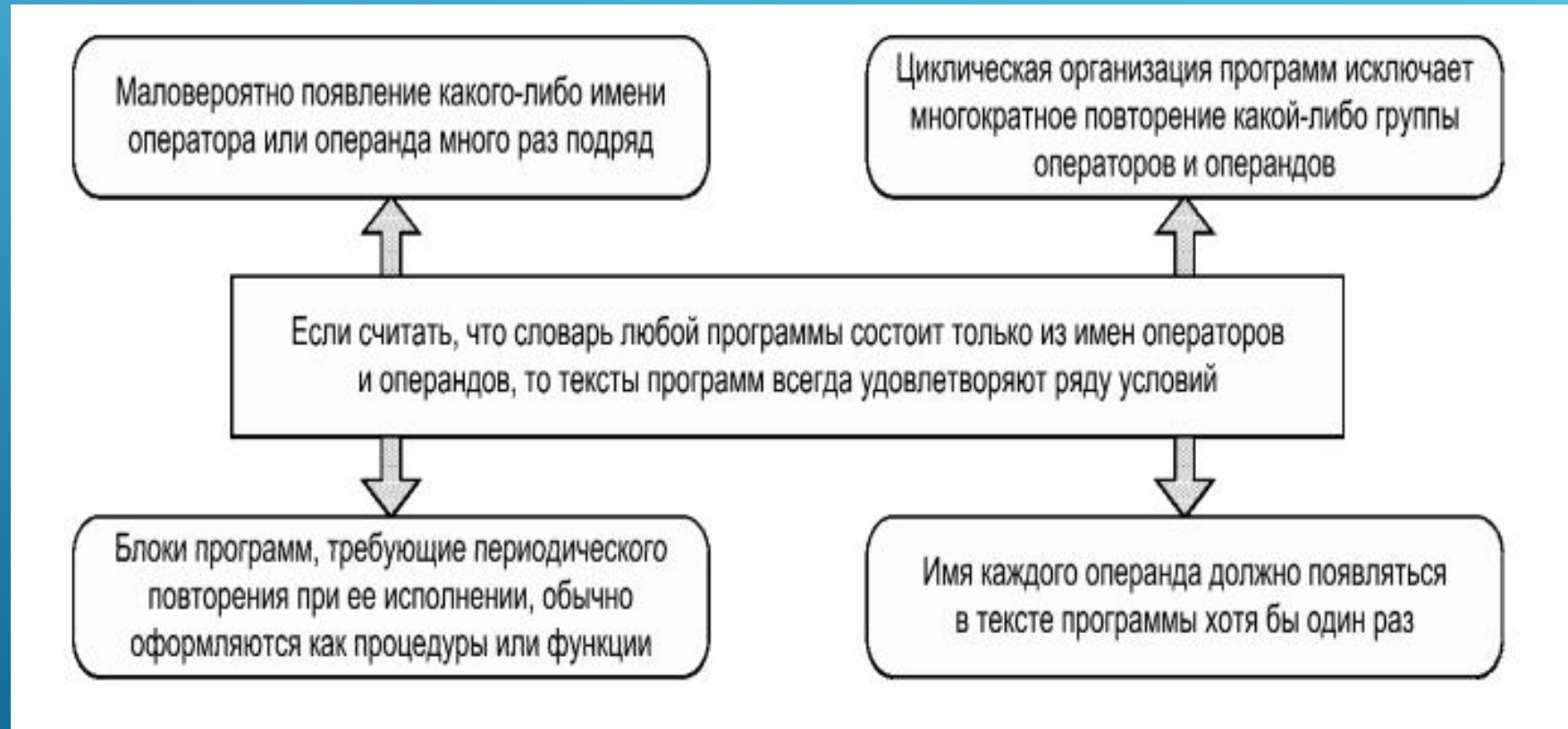
Абдурахманова Н.Н.

ОСОБЕННОСТИ ФОРМИРОВАНИЯ СЛОВАРЯ ПРОГРАММЫ

Любая программа определяет последовательность действий над операндами с помощью операторов. Исходный текст программы, записываемой на том или ином языке программирования представляет собой набор текстовых строк, которые записываются по специальным правилам, и в том числе имеет свои элементы. Операнд – это некоторый объект или величина, обрабатываемая в программе, а оператор представляет собой обозначение конкретного действия, выполняемого по отношению к операнду.

- ▶ Если считать, что словарь любой программы состоит только из имен операторов и операндов, то тексты программ всегда удовлетворяют следующим условиям:
 - маловероятно появление какого-либо имени оператора или операнда много раз подряд – языки программирования, как правило, позволяют создавать такие конструкции, в которых подобные фрагменты программы имеют минимальную длину;
 - циклическая организация программ исключает многократное повторение какой-либо группы операторов и операндов – более компактные варианты текстов получаются при разумном использовании развитых возможностей языков программирования, причем многообразие языков предоставляет богатую палитру инструментов;
 - блоки программ, требующие периодического повторения при ее исполнении, обычно оформляются как процедуры или функции, поэтому в текстах программ достаточно применения только их имен;
 - имя каждого операнда должно появляться в тексте программы хотя бы один раз – многие среды программирования обращают внимание программистов на неиспользуемые имена, которые следует удалять из текста программ, чтобы сократить объем памяти, используемой при объявлении переменных.

СХЕМАТИЧНО ЭТИ УСЛОВИЯ, ОТНОСЯЩИЕСЯ К ТЕКСТУ ПРОГРАММ, ОТОБРАЖЕНЫ НА РИС.



ИЗМЕРЯЕМЫЕ СВОЙСТВА ПРОГРАММ

- ▶ В состав измеримых свойств любого представления алгоритма (или программы) могут быть включены следующие метрические характеристики:
 - η_1 – число простых (уникальных) операторов, появляющихся в данной реализации;
 - η_2 – число простых (уникальных) операндов, появляющихся в данной реализации;
 - N_1 – общее число всех операторов, появляющихся в данной реализации;
 - N_2 – общее число всех операндов, появляющихся в данной реализации;
 - f_{1j} – число появлений в программе j -го оператора, где $j = 1, 2, 3, \dots, \eta_1$;
 - f_{2j} – число появлений в программе j -го операнда, где $j = 1, 2, 3, \dots, \eta_2$

- ▶ η_2 . Учитывая эти основные метрические характеристики для программы, в конкретной реализации текста программы можно определить:
 - словарь $\eta = \eta_1 + \eta_2$;
 - длину реализации программы $N = N_1 + N_2$;
 - длину программы $\tilde{N} = (\eta_1 \cdot \log_2 \eta_1) + (\eta_2 \cdot \log_2 \eta_2)$.

МЕТРИКИ ХОЛСТЕДА:

- ▶ ВЫЧИСЛЯЮТСЯ НА ОСНОВАНИИ АНАЛИЗА ЧИСЛА СТРОК И СИНТАКСИЧЕСКИХ ЭЛЕМЕНТОВ ИСХОДНОГО КОДА ПРОГРАММЫ



- ▶ М. Холстед вводит аппроксимирующую формулу для измерения теоретической длины программы - N^{\wedge}

$$N^{\wedge} = n_1 * \log_2(n_1) + n_2 * \log_2(n_2), (10)$$

где n_1 - словарь операторов;

n_2 - словарь операндов программы.

► Вводя эту оценку, Холстед исходит из основных концепций теории информации, по аналогии с которыми частота использования операторов и операндов в программе пропорциональна двоичному логарифму количества их типов. Таким образом, выражение (10) представляет собой идеализированную аппроксимацию (1), т. е. справедливо для потенциально корректных программ, свободных от избыточности или несовершенств (стилистических ошибок). несовершенствами можно считать следующие ситуации:

- последующая операция уничтожает результаты предыдущей без их использования;
- присутствуют тождественные выражения, решающие совершенно одинаковые задачи;
- одной и той же переменной назначаются различные имена и т. П.

- ▶ Подобные ситуации приводят к изменению N без изменения n .
- ▶ М. Холстед утверждает, что для стилистически корректных программ отклонение в оценке теоретической длины N^\wedge от реальной N не превышает 10%.
- ▶ Значение N^\wedge может быть использовано как эталонное значение длины программы со словарем n . Длина корректно составленной программы N , т. е. программы, свободной от избыточности и имеющей словарь n , не должна отклоняться от теоретической длины программы N^\wedge более чем на 10%. Таким образом, измеряя n_1, n_2, N_1 и N_2 и сопоставляя значения N и N^\wedge для некоторой программы, при более чем 10%-ном отклонении можно говорить о наличии в программе стилистических ошибок, т. е. несовершенств.

- ▶ На практике N и N^\wedge часто существенно различаются.
- ▶ М. Холстед вводит n^* – теоретический словарь программы, т.е. словарный запас, необходимый для написания программы, с учетом того, что необходимая функция уже реализована в данном языке и, следовательно, программа сводится к вызову этой функции. Например, согласно М. Холстеду, возможное осуществление процедуры выделения простого числа могло бы выглядеть так:

- ▶ $CALL\ SIMPLE(X, Y)$,
- ▶ где Y – массив численных значений, содержащий искомое число X .
- ▶ Теоретический словарь в этом случае будет состоять из
- ▶ $n_1^* : \{CALL, SIMPLE(\dots)\}, n_1^* = 2;$
- ▶ $n_2^* : \{X, Y\}, n_2^* = 2,$
- ▶ а его длина, определяемая как
- ▶ $n^* = n_1^* + n_2^*,$
- ▶ будет равняться 4.
- ▶ Используя n^* , Холстед вводит оценку V^*
- ▶ $V^* = n^* * \log_2 n^*, (11)$

МЕТРИКИ ХОЛСТЕДА

- ▶ Основу метрики Холстеда составляют четыре измеряемые характеристики программы:
- ▶ NUOprtr (Number of Unique Operators) — число уникальных операторов программы, включая символыразделители, имена процедур и знаки операций (словарь операторов);
- ▶ NUOprnd (Number of Unique Operands) — число уникальных операндов программы (словарь операндов);
- ▶ Noprtr (Number of Operators) — общее число операторов в программе;
- ▶ Noprnd (Number of Operands) — общее число операндов в программе.

ОЦЕНКИ НА ОСНОВЕ МЕТРИКИ ХОЛСТЕДА

- ▶ **Словарь программы** (Halstead Program Vocabulary):

$$HPVoc = NUOprtr + NUOprnd;$$

- ▶ **Длина программы** (Halstead Program Length):

$$HPLen = Noprtr + Noprnd;$$

- ▶ **Объем программы** (Halstead Program Volume):

$$HPVol = HPLen \log_2 HPVoc;$$

- ▶ **Сложность программы** (Halstead Difficulty, HDiff):

$$HDiff = (NUOprtr/2) \times (NOprnd / NUOprnd);$$

- ▶ **На основе показателя HDiff** предлагается оценивать усилия программиста при разработке при помощи показателя HEff (Halstead Effort):

$$HEff = HDiff \times HPVol.$$

- ▶ Также на основе показателя $NDiff$ имеется возможность оценивать усилия программиста при разработке при помощи показателя $NEff$ (Halstead Effort): $NEff = NDiff \times NPVol$.

МЕТРИКИ СЛОЖНОСТИ ПОТОКА УПРАВЛЕНИЯ ПО

- ▶ Метрики второй группы базируются на анализе управляющего графа программы. Управляющий граф $G(V, E)$ строится в виде ориентированного графа, в котором вычислительные операторы или выражения представляются в виде вершин $V(V_h \dots, V_m)$, а передача управления между вершинами — в виде ребер $E(E_x, E_p)$. Как правило, на основании анализа данного графа строят метрики для оценки количества управляющих переходов внутри программ, либо взаимосвязями этих переходов.