МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Реферат з предмету «Економіка організації підприємства»

На тему: «Формули Холстеда»

Виконав:

Студент групи КН-36a

Жураковський Д. Г.

Перевірила:

Єршова С. І.

Харків 2020

Формули Холстеда відноситься до формул, що обчислюється на підставі аналізу числа рядків і синтаксичних елементів вихідного коду програми.

Основу формул Холстеда складають чотири вимірювані характеристики програми:

* NUOprtr (Number of Unique Operators) - число унікальних операторів програми, включаючи символи-роздільники, імена процедур і знаки операцій (словник операторів);
* NUOprnd (Number of Unique Operands) - число унікальних операндів програми (словник операндів);
* Noprtr (Number of Operators) - загальне число операторів в програмі;
* Noprnd (Number of Operands) - загальне число операндів в програмі.

Формули Холстеда відображають лексичний підхід до вимірювання характеристик програмного забезпечення, заснований на вимірних властивості алгоритмів. Властивості будь-якого опису алгоритму (або програми для ЕОМ), на думку Холстеда, можуть бути виміряні або обчислені на основі наступних метричних характеристик (оціночних елементів):

* n1 - кількість різних операторів програми;
* n2 - кількість різних операндів програми;
* N1 - загальна кількість операторів програми;
* N2 - загальна кількість операндів програми.

На їх основі Голстед визначає наступні формули:

1. словник програми (в умовних одиницях)

n = n1+n2, (1)

1. теоретичний словник програми, тобто словниковий запас, необхідний для написання програми, з урахуванням того, що необхідна формула вже реалізована в даній мові і, отже, програма зводиться до виклику цієї формули.

n\* = n1\* + n2\*

1. довжина програми (в умовних одиницях)

N = N1 + N2 (2)

1. теоретична довжина програми (в умовних одиницях)

Ñ = (n1 \* log2 n1) + (n2 \* log2 \* n2), (3)

Вводячи цю оцінку, Голстед виходить з основних концепцій теорії інформації, за аналогією з якими частота використання операторів і операндів в програмі пропорційна бінарного логарифму кількості їх типів. Таким чином, вираз (3) являє собою ідеалізовану апроксимацію (2), т. Е. Справедливо для потенційно коректних програм, вільних від надмірності або недосконалостей (стилістичних помилок).

Недосконалостями можна вважати наступні ситуації:

* подальша операція знищує результати попередньої без їх використання;
* присутні тотожні вирази, які вирішують абсолютно однакові завдання;
* однією і тією ж змінної призначаються різні імена і т. П.

Подібні ситуації призводять до зміни N без зміни n.

Голстед стверджує, що для стилістично коректних програм відхилення в оцінці теоретичної довжини Ñ від реальної N не перевищує 10%.

Пропонується використовувати Ñ як еталонне значення довжини програми зі словником n. Довжина коректно складеної програми N, т. Е. Програми, вільної від надмірності і має словник n, не повинна відхилятися від теоретичної довжини програми Ñ більш ніж на 10%. Таким чином, вимірюючи n1, n2, N1 і N2 і зіставляючи значення N і Ñ для деякої програми, при більш ніж 10% -не відхилення можна говорити про наявність в програмі стилістичних помилок, т. Е. Недосконалостей.

На практиці N і Ñ часто істотно відрізняються.

* обсяг програми (в бітах, визначення обсягу програми передбачає, що позначення операндів і операторів вимагає не більше одного символу).

V = N × log2(n) (4)

* потенційний обсяг програми (вважаючи, що в програмах, ідеальних з точки зору економії витрат пам'яті, по-перше: оператори і операнди не повторюються, по-друге: все операнди є або вхідними, або вихідними параметрами і по-третє: для запису тексту програми достатньо двох операторів (опису заголовка процедури-функції і присвоювання значення))

V\* = n\* × log2 n\* (5),

де n2\* - загальне число вхідних і вихідних параметрів.

* рівень якості програми (в умовних одиницях, якщо n2\* = 2)

L = V\*/ V (6)

Вихідним для введення цієї характеристики є припущення про те, що при зниженні стилістичного якості програмування зменшується змістовне навантаження на кожен компонент програми і, як наслідок, розширюється обсяг реалізації вихідного алгоритму. З огляду на це, можна оцінити якість програмування на підставі ступеня розширення тексту щодо потенційного обсягу V \*. Очевидно, для ідеальної програми L = 1, а для реальної - завжди L <1.

Нерідко доцільно визначити рівень програми, не вдаючись до оцінки її теоретичного обсягу, оскільки список параметрів програми часто залежить від реалізації і може бути штучно розширено. Це призводить до збільшення метричної характеристики якості програмування. М. Холстед пропонує апроксимувати цю оцінку виразом, що включає тільки фактичні параметри, т. Е. Параметри реальної програми:

L\* = 2 \* n2 / (n1 \* N2)

* рівень мови - це коефіцієнт пропорційності зміни обсягу програми при перекладі з однієї мови на іншу так, що забезпечується сталість твори рівня програми на потенційний обсяг.

λ=L × V\* (7)

* інтелектуальний зміст програми (інваріантне по відношенню до використовуваних мов реалізації, в умовних одиницях)

I=L\*× V (8)

Перетворюючи вираження (8) з урахуванням (6), отримуємо

I = L\*× V = L×V = V\*×V/V = V\*.

Еквівалентність I і V\* свідчить про те, що ми маємо справу з характеристикою інформативності програми.

Введення характеристики I дозволяє визначити розумові витрати на створення програми. Процес створення програми умовно можна уявити як ряд операцій:

1. осмислення пропозиції відомого алгоритму;
2. запис пропозиції алгоритму в термінах використовуваної мови програмування, т. е. пошук в словнику мови відповідної інструкції, її змістове наповнення і запис.

Використовуючи цю формалізацію в методиці Холстеда, можна сказати, що написання програми по заздалегідь відомим алгоритмом є N^ - кратноє вибірка операторів і операндів зі словника програми n, причому число порівнянь (за аналогією з алгоритмами сортування) складе log2(n).

Якщо врахувати, що кожна вибірка-порівняння містить, в свою чергу, ряд уявних елементарних рішень, то можна поставити у відповідність змістовної навантаженні кожної конструкції програми складність і число цих елементарних рішень. Кількісно це можна характеризувати за допомогою характеристики L, оскільки 1 / L має сенс розглядати як середній коефіцієнт складності, що впливає на швидкість вибірки для даної програми. Отже можна розрахувати оцінку необхідних інтелектуальних зусиль.

* оцінка інтелектуальних зусиль (в умовних одиницях)

E = N\*× log2 (n / L) (9)

E характеризує число необхідних елементарних рішень при написанні програми.

Однак слід зауважити, що E адекватно характеризує лише початкові зусилля з написання програм, оскільки при побудові E не враховуються налагоджувальні роботи, які вимагають інтелектуальних затрат іншого характеру.

Суть інтерпретації цієї характеристики полягає в оцінці не витрат на розробку програми, а витрат на сприйняття готової програми. При цьому замість теоретичної довжини програми N\* використовується її реальна довжина:

E\* = N × log2(n / L)

Характеристика E \* введена, виходячи з припущення, що інтелектуальні зусилля на написання і сприйняття програми дуже близькі за своєю природою. Однак якщо при написанні програми стилістичні похибки в тексті практично не мають відображатися на інтелектуальної трудомісткості процесу, то при спробі зрозуміти таку програму їх присутність може привести до серйозних ускладнень. Ця посилка досить добре узгоджується з нашими висновками щодо взаємозв'язку N і N ^, викладеними вище.

Перетворюючи формулу (9) з урахуванням виразів (4) і (6), отримуємо

E = V × V / V\*

Таке уявлення E', а відповідно і E, так як E = E', наочно ілюструє доцільність розбиття програм на окремі модулі, оскільки інтелектуальні витрати виявляються пропорційними квадрату обсягу програми, який завжди більше суми квадратів обсягів окремих модулів.

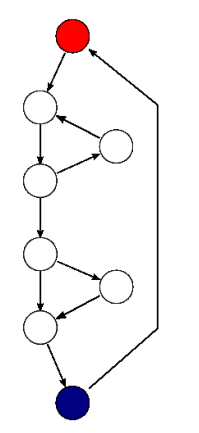
* час на програмування (в умовних одиницях)

T = E / S, (10) или Т ≅ (n1 × N2 × log2n × (n1 × log2n1 + n2 × log2n2)) /   
(2 × n2 × S), (11)

де S - число Страуда (5 < S < 20).

Число Страуда Голстед прийняв рівним 18 - число розумових операцій в одиницю часу.

Граф потоку управління програми. Графом потоку управління програми називається орієнтований граф, в вузлах якого знаходяться неподільні групи команд, а ребрами з'єднуються такі блоки, які можуть бути виконані відразу один за одним.



Потенційний обсяг програми є мірою мінімально необхідного обсягу програми з заданим словником. При цьому потенційний обсяг не залежить від мови реалізації. При перекладі програми з однієї мови на інший потенційний обсяг не змінюється, але дійсний обсяг V або збільшується, або зменшується в залежності від мови реалізації.

Використовуючи вираз для потенційного обсягу програми, Холстедом отримані наступні функції:

Число Страуда S визначається як число «страудовскіх моментів» в секунду. «Страудовскій момент» - це час, необхідний людині для виконання елементарного розрізнення об'єктів, подібно розрізнення кадрів фільму. Страуд виявив, що людина здатна розрізняти від 5 до 20 об'єктів в секунду.

Рівень програми L < 1 характеризує ефективність реалізації алгоритму щодо витрат пам'яті. Тільки для найбільш стислій форми реалізації алгоритму (V = V\*) рівень програми має значення 1. Всім іншим варіантам реалізації відповідають значення L < 1.

Інтелектуальний зміст характеризує міру «сказаного» в програмі, або її «інформативність». Інтелектуальний зміст (рівень) програми сильно корелює з потенційним обсягом (L = V\*) і теж не залежить від мови реалізації.

Робота з програмування (рівняння уявної роботи) характеризує величину розумової роботи, пов'язаної з написанням програмного коду. Так як сума квадратів двох величин завжди менше квадрата їх суми, рівняння роботи дає підставу для розбиття програми на складові частини - модулі. Модульність знижує роботу з програмування. Дослідження можливостей оперативного мислення людини дають підстави вважати, що найбільш продуктивна ситуація, при якій для отримання одного результату використовується не більше п'яти об'єктів. У прикладному відношенні цей результат називають гіпотезою про «шести об'єктах». Для визначення кількості модулів M в програмі Голстед рекомендує використовувати вираз

M= n2\*/6, (12)

де n2\* - загальна кількість вхідних і вихідних змінних в програмі.

З рівняння роботи отримано наступне рівняння помилок в програмі

B=L × E/E0, (13)

де В - кількість помилок в програмі, Е0 - середнє число елементарних відмінностей між можливими помилками програмування.

Використовуючи перетворене рівняння роботи

Е= (V\*)3/λ2, (14)

значення рівня англійської мови (λ = 2,16) в якості аналога мови програмування і гіпотезу про «шести об'єктах» ідеальної за витратами пам'яті програми (n1 = n1\* = 2, n2 = n2\* = 6), Голстед вивів наступне рівняння для прогнозу кількості помилок в програмі:

В=V/3000, (15)

где V - объем программы.

**Висновок**

Формули Холстеда відноситься до формул, що обчислюється на підставі аналізу числа рядків і синтаксичних елементів вихідного коду програми. Формули Холстеда відображають лексичний підхід до вимірювання характеристик програмного забезпечення, заснований на вимірних властивості алгоритмів. Властивості будь-якого опису алгоритму (або програми для ЕОМ), на думку Холстеда, можуть бути виміряні або обчислені на основі наступних метричних характеристик (оціночних елементів)

**Список використаних джерел:**

1. [https://glebradchenko.susu.ru/courses/bachelor/engineering] – Кодирование
2. [https://opnsc.ru/metriki-holsteda.html] - Формулы Холстеда
3. [https://works.doklad.ru/view/v2-\_U6AoJs4/6.html] - Критерии качества программного обеспечения