ТЕЗИС ЧЁРЧА. АЛГОРИТМИЧЕСКИ НЕРАЗРЕШИМЫЕ ПРОБЛЕМЫ. ПРЕОБРАЗОВАНИЕ СХЕМ АЛГОРИТМОВ

ПЛАН ЛЕКЦИИ

1. Тезис Чёрча
2. Неразрешимые алгоритмические проблемы
3. Понятие частичного алгоритма
4. Алгоритм преобразования структурных схем алгоритмов Ашкрофта-Манны

ТЕЗИС ЧЁРЧА (1)

Формулировка 1.

Числовая функция тогда и только тогда алгоритмически (машинно) вычислима, когда она частично рекурсивна.

Church Alonzo. An unsolvable problem of elementary number theory // Amer. J. math. – 1963. v. 58 – N 2. – p. 345 – 363.

Пост и Тьюринг одновременно с Чёрчем и не зависимо друг от друга утверждают, что

класс всюду определённых функций, вычислимых в определённой модели, совпадает с классом всюду определённых вычислимых функций (для фиксированных ансамблей).

С. Клини уточнил, что тезис Чёрча, в узком смысле, утверждает, что

всякая вычислимая функция с натуральными аргументами и значениями частично рекурсивна.

Kleene Stephen. Recursive functions and intuitionistic mathematics // Proceedings of the International congress of mathematicians, Cambridge (Massachusetts), USA, August 30 – September 6, 1950. – v. 1. Providence: AMS. – p. 679 – 685.

ТЕЗИС ЧЁРЧА (2)

Kleene Stephen. Introduction to the mathematics. – N.Y.; Toronto: D. Van Nostrand Company, 1952. – 516 p.

По этой причине тезис называют тезисом Чёрча – Тьюринга – Поста – Клини.

Формулировка 2.

Для любой функции $f \in$ Computable(Σ^*, Σ^*) существует алфавит Г и машина Тьюринга с алфавитом $\Gamma \supseteq \Sigma^*$, которая вычисляет f.

Формулировка 3.

Класс алгоритмически (машинно) вычислимых частичных числовых функций совпадает с классом всех частично рекурсивных функций.

(!!!) Все формальные уточнения понятия вычислимости не вывели из класса функций, вычислимых по Тьюрингу

АЛГОРИТМИЧЕСКИ НЕРАЗРЕШИМЫЕ ПРОБЛЕМЫ (1)

Д. Гильберт: “В математике не может быть неразрешимых проблем”, конгресс 1900 г. в Париже.

Теорема.

Не существует алгоритма (машины Тьюринга), позволяющего по описанию произвольного алгоритма и его исходных данных (и алгоритм и данные заданы символами на ленте машины Тьюринга) определить, останавливается ли этот алгоритм на этих данных или работает бесконечно.

ПРИЧИНЫ АЛГОРИТМИЧЕСКОЙ НЕРАЗРЕШИМОСТИ И ПРОБЛЕМЫ

I. *Отсутствие общего метода решения задачи*

Примеры.

1. Распределение девяток в записи числа π.

Исходим из существования функции $f(n) = i$, где n – количество девяток подряд в десятичной записи числа $\pi=3,141592\ldots$, а i – номер самой левой девятки из n девяток подряд. Для $n = 1$ функция $f(1) = 5$.

АЛГОРИТМИЧЕСКИ НЕРАЗРЕШИМЫЕ ПРОБЛЕМЫ (2)

2. Вычисление совершенных чисел (это числа, которые равны сумме своих делителей, например: 28 = 1+2+4+7+14).

   Пусть существует функция $S(n)$ = n-ое по счёту совершенное число. Стоит задача вычисления $S(n)$ по произвольно заданному n.

3. Десятая проблема Гильберта

   Имеется многочлен n-ой степени с целыми коэффициентами. Необходимо выяснить, существует ли у данного уравнения целочисленное решение?

   Ю.В. Матиясевич в 1970 г. Доказал, что алгоритм решения проблемы не существует.

4. При этих же условиях выяснить, существует ли решение в рациональных числах?

   Факт существования не доказан и не опровергнут.

II. *Информационная неопределённость задачи*

Пример.

1. Позиционирование машины Поста на последний помеченный ящик

АЛГОРИТМИЧЕСКИ НЕРАЗРЕШИМЫЕ ПРОБЛЕМЫ (3)

2. Игра эволюция: по произвольной позиции нельзя узнать, вымрет колония или нет

III. *Логическая неразрешимость в смысле теоремы Гёделя о неполноте*

Примеры

1. Проблема останова (теорема выше).
2. Проблема эквивалентности алгоритмов.

   По двум произвольным заданным алгоритмам (например, по двум машинам Тьюринга) определить, будут ли они выдавать одинаковые выходные результаты на любых исходных данных.

3. Проблема тотальности

   По произвольному заданному алгоритму определить, будет ли он останавливаться на всех возможных наборах исходных данных. В другой формулировке: является ли частичный алгоритм P всюду определённым?

АЛГОРИТМИЧЕСКИ НЕРАЗРЕШИМЫЕ ПРОБЛЕМЫ (4)

Теорема Гёделя

Если формальная система S непротиворечива, то формула A невыводима в S; если система S ω-непротиворечива, то формула ¬A невыводима в S. Таким образом, если система S ω-непротиворечива, то она неполна и A служит примером неразрешимой формулы.

ПОНЯТИЕ ЧАСТИЧНОГО АЛГОРИТМА (1)

В. Пост (1943 г.) сформулировал проблему.

Задан алфавит Σ, $|\Sigma| \geq 2$ и конечное множество непустых пар, определённых как $(\Sigma^+, \Sigma^+) - (x_i, y_i)$, $x_i \in \Sigma^+$, $y_i \in \Sigma^+$, $i = 1, \ldots, |\Sigma^+|$, $j = 1, \ldots, |\Sigma^+|$.

Найти *решающую* последовательность, паче такая найдётся.

Решающая последовательность – конечная последовательность пар, (не обязательно из различных x_i и y_i), такая, что цепочка, *составленная из левых* компонентов пар, совпадает с последовательностью, *составленной из правых подцепочек*.

Пример

Задан $\Sigma = \{0, 1\}$ – алфавит. Вход алгоритма: пары {(0111, 1); (0, 001); (10, 1)}.

Решающая последовательность такова: (**0**, *001*), (**0**, *001*), (**10**, *1*), (**0111**, *1*), ибо

0 **0** **10** **0111** ≡ *001* *001* *1* *1*

Для входа (01, 010), (010, 100), (100, 00) решения не будет.

Какие бы пары цепочек не брали (010, 100), (100, 00) – не совпадают головы, а (01, 010) – не достичь соответствия количества нулей.

ПОНЯТИЕ ЧАСТИЧНОГО АЛГОРИТМА (2)

Для решения задачи можно сконструировать *частичный* алгоритм, строящий всевозможные упорядоченные возможные последовательности, проверяющий для каждой генерации условия решения.

Частичный алгоритм возможно, но *не обязательно* (**!**), находит решение проблемы.

Если существует решающая последовательность, то решение может быть получено *за конечное число шагов.*

Поскольку ***имеется частичный алгоритм***, то задача ***называется частично-разрешимой***.

Так как *общий метод* определения отсутствия решающей последовательности *не может быть указан*, следовательно, ***задача сведена к проблеме останова***, следовательно, ***алгоритмически не разрешима***.

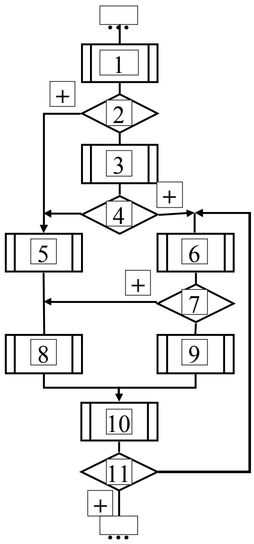
Алгоритм нахождения эквивалентности и тотальности не имеют даже частичных алгоритмов

Преобразование структурных схем алгоритмов (1)

Achcroft E., Manna Z.

The Translation of ‘GO TO’ Programs to ‘WHILE’ Programs., Proc. IFIP Cong., North-Holland Publ. Co., Amsterdam, p.p. 250 – 255, 1971.

Преобразование структурных схем алгоритмов (2)




```
NumBock := 1;
while (NumBock != 0)
     switch (NumBock)
            case 1
            …
            NumBock := 2;
            case 2
            if (…) then NumBock := 5;
                     else NumBock := 3;
            …
             case 10 …
            NumBock := 11;
            case 11
            if (…) then NumBock := 0;
                     else NumBock := 6;
            otherwise print("Error");
     end switch;
end while;
```