Домашнее задание к Лекции 1:
«Теория формальных языков и трансляций»
Пятый семестр.

Специальность 02.03.03.
Математическое обеспечение и администрирование информационных систем.
Кафедра системного программирования.
Преподаватель - Федорченко Людмила Николаевна.
Группа 344.
Санкт-Петербург 2020.

Фомина В.В.

Набрано в LaTeX

Дата изменения: 10 сентября 2020 г. 09:28

1

Задача 1.1.

Функция $K(i,j) = \dfrac{(i+j-1)(i+j-2)}{2} + j$ отображает упорядоченные пары целых на целые. Найти обратные функции $I(K)$ и $J(K)$ с таким свойством, что: $I(K(i,j)) = i$ и $J(K(i,j)) = j$.

Составьте процедуру на Паскале, которая по целому $k > 0$ выдаёт i и j — номер строчки и столбца треугольной сетки, где расположено значение k.

Решение:

```
var
  i, j, k, numberOfElementsInTriangle, numberOfLevels : integer;

begin
  readln(k);

  numberOfLevels := 0;

  while numberOfLevels * (numberOfLevels + 1) div 2 < k do
    numberOfLevels += 1;

  numberOfLevels -= 1;

  numberOfElementsInTriangle := numberOfLevels * (numberOfLevels + 1) div 2;

  j := k - numberOfElementsInTriangle;
  i := numberOfLevels - j + 2;

  writeln('i = ', i);
  writeln('j = ', j);

end.
```

Рис. 1: Решение

Тесты:

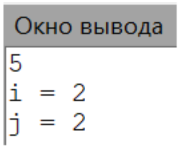

Рис. 2: Тест 1

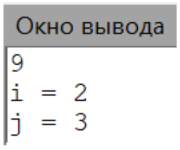

Рис. 3: Тест 2

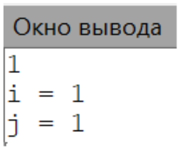

Рис. 4: Тест 3

Задача 1.2.

Пусть $\hat{J}\,(w, x, y) = J(w, J(x, y))$.

Какая тройка целых w, x, y приписывается числу 1000, если

$J(x, y) = \dfrac{(x + y - 1)(x + y - 2)}{2} + y$.

Решение:

$\hat{J}\,(w, x, y) = J(w, J(x, y)) = 1000 := k$

Применим процедуру из **Задачи 1** к k и найдем $J(x, y)$ и w.

Получим $J(x, y) = 10 := t,\ w = 36$.

Теперь применим процедуру из **Задачи 1** к t и найдем x и y.

Получим $x = 1,\ y = 4$.

Ответ: $(36, 1, 4)$.

Задача 1.3.

Опишите простую процедуру для перенумерации предложений рекурсивного языка.

Решение:

Определение 1.1.

Язык рекурсивен, если существует алгоритм его распознавания.

Определение 1.2.

Алгоритм распознавания – алгоритм, определяющий есть ли данное предложение в данном языке или нет.

V – алфавит, L – язык, V^* – множество всех предложений, состоящих из символов алфавита.

Пусть в алфавите V p символов. Пронумеруем их числами от 0 до $p-1$. Поставим предложениям из V^*, а также предложению ε в соответсвие числа из p-ичной системы счисления. Реализуем это следующим образом. Предложению ε поставим в соответсвие число 0. Пронумеруем предложения из V^* в порядке увеличения длины, причём все предложения одинаковой длины будем нумеровать в соответствии с лексикографическим порядком, определенным нумерацией символов алфавита.

В соответствии с нумерацией будем отправлять предложения (ε и предложения из V^*) на проверку принадлежности языку L через алгоритм распознавания. Если предложение принадлежит языку, то ставим ему в соответсвие число k ($k = 1$ изначально) и увеличиваем k на 1.

Задача 1.4.

Докажите, что если существует процедура для перечисления множества целых в монотонном порядке, то это множество рекурсивно в том смысле, что существует алгоритм определения, находится ли данное целое в этом множестве.

Решение:

Хотим определить принадлежит ли число k множеству. Запускаем процедуру перечисления. Если текущее число, выдаваемое процедурой

- меньше k, смотрим дальше;
- равно k – да, число k принадлежит множеству, останавливаем процедуру;
- больше k – нет, число k множеству не принадлежит, останавливаем процедуру.

Рассмотрим последний случай. Процедура перечисления закончила свою работу и последнее число, которое она выдала меньше k. В таком случае число k множеству не принадлежит.
Таким образом мы построили алгоритм определения, находится ли произвольное целое в множестве.

Задача 1.5.
Покажите, что все конечные множества рекурсивны.

Решение:

Пусть k – элемент, принадлежность множеству которого мы хотим проверить. Множество конечно $\Rightarrow$ можно построить биекцию между элементами множества и целыми числами из отрезка $[1..n]$, где n – мощность множества. В соответсвии с полученной нумерацией реализуем процедуру перечисления. Запустим ее. Если текущий элемент, выдаваемый процедурой перечисления совпадает с k, то k принадлежит множеству, останавливаем процедуру. Если процедура завершила свое выполнение, а про принадлежность элемента k множеству не было получено никакой информации, значит k множеству не принадлежит.