**Рекурсії 2-го рівня**

Як уже зазначалося, функція *f* називається рекурсивною, якщо вона може бути обчислена алгоритмом

function *f*(*x*1, … , *xn*, *y*)

begin

if *y* = 0 then *f* := *g*(*x*1, … , *xn*)

else *f* := *h*(*x*1, … , *xn*, *y*, *f*(*x*1, … , *xn*, *y* – 1))

end,

тобто, значення такої функції обчислюються через значення цієїж функції для менших значень аргументу. Якщо відоме значення функції для *y* = 0, то за допомогою приведеного алгоритму можна обчислити значення функції для будь-якого *y*. Крім того, при умові, що *g*, *h* є ПРФ, функція *f* буде ПРФ.

Функція *f* виникає із функції *h* рекурсією з поверненням, якщо вона може бути обчислена алгоритмом

function *f*(*x*)

begin

if *x* = 0 then *f* := *b*

else *f* := *h*(*x*, *f*(*α*(*x*)))

end.

При умові, що *h* є ПРФ, функція *f* буде ПРФ, тобто може бути обчислена алгоритмом

function *f*(*x*)

begin

if *x* = 0 then *f* := *b*

else *f* := *G*(*x*, *f*(*x* – 1))

end.

Значення такої функції обчислюються через значення цієїж функції для менших значень аргументу (*α*(*x* + 1) ≤ *x*).

У кожному з вищеприведених алгоритмів функція обчислюється рекурсією по одному аргументу. Для того, щоб задати функцію рекурсією по двох аргументах (обчислювати її значення через її ж значення, але для менших значень аргумента), треба упорядкувати множину пар натуральних чисел, наприклад в такий спосіб

(0,0) < (0,1) < … < (1,0) < (1,1) < … < (2,0) < (2,1) < … ,

а потім задати алгоритм визначення значення функції через значення цієї ж фунції для менших значень аргументів. Наприклад, обчислення значення функції *F*(*x*, *y*) рекурсією по двох аргументах полягає у заданні значеня *F*(0, 0) та у визначенні *F*(*x*, *y*) через значення цієї функції для менших значень агументу. Можливі випадки:

а) Для *F*(0, *x*) значення з меншим аргументом має вигляд *F*(0, *y*), де *y* < *x*.

b) Для *F*(*n* + 1, 0) значення з меншим аргументом має вигляд *F*(*m*, *x*), де *m* ≤ *n*, а *x* довільне.

c) Для *F*(*n* + 1, *x* + 1) значення з меншим аргументом мають вигляд *F*(*n +* 1, *x*), *F*(*n*, *g*(*x*)), *F*(*n*, *F*(*n* + 1, *x*)), … .

**Приклад.** Нехай функція *B*(*n*, *x*) визначається схемою

*B*(*n* + 1, *x* + 1) = *B*(*n*, *B*(*n* + 1, *x*)),

*B*(*n* + 1, 0) = *B*(*n*, 1),

*B*(0, *x*) = 1 + *x*.

Ця функція задається рекурсією 2-го рівня, оскільки значення функції *B*(*n*, *x*) обчислюється через значення цієї ж фунції для менших значень аргументів. Зрозуміло, що така функція обчислюється алгоритмом

function *B*(*n*, *x*)

begin

if *n* ≥ 1 ∧ *x* = 0 then *B* := *B*(*n* – 1, 1)

if *n* = 0 then *B* := 1 + *x*

if *x* ≥ 1 ∧ *n* ≥ 1 then *B* := *B*(*n* – 1, *B*(*n*, *x* – 1))

end.

Якщо всі функції, які зустрічаються в цьому алгоритмі є ПРФ, то функція *B*(*n*, *x*) буде по крайній мірі рекурсивною функцією.

**Універсальні функції**

Нехай *ℑ* –система часткових одномісних функцій.

Часткова функція *F*(*x*, *y*) від двох змінних називається універсальною для сімейства *ℑ*, якщо виконуються наступні умови:

1. Для кожного фіксованого *i* функція *F*(*i*, *y*) належить *ℑ*;

2. Для кожної функції *f*(*y*) із *ℑ* існує таке число *i*, що для всіх *y* *F*(*i*, *y*) = *f*(*y*).

**10.1. Універсальна рекурсивна функція**

**Теорема 10.1.** Система всіх одномісних ПР функцій має універсальну рекурсивну функцію.

Така функція позначається через *D*(*n*, *x*) і має наступні властивості:

1. Для кожного фіксованого  одномісна функція *D*(*n*, *x*) є ПР функцією.

2. Для кожної одномісної ПР функції *f*(*x*) існує число *n* таке, що *D*(*n*, *x*) = *f*(*x*).

**Наслідок.** Функція *Dn*+1(*x*0, *x*1, … , *xn*) = *D*(*x*0, ***c***(*x*1, … , *xn*)) є рекурсивною функцією універсальною для класу *n*-місних ПРФ .

**10.2. Універсальна ЧРФ**

**Теорема 10.2.** Кожна ЧРФ *f*(*x*1, … , *xn*) може бути обчислена алгоритмом:

function *f* (*x*1, ... , *xn*)

begin

*i* := 0

while *F*(*x*1, ... , *xn*, *i*) ≠ 0 do

*i* := *i* +1

*f* := *G*(*i*)

end,

де *F*, *G* деякі (залежні від *f*) ПРФ.

Доведення. Якщо *f* – ЧРФ, то її графік *Gf* є РПМ. Тому *Gf* = <*f*1(*x*), … , *fn*(*x*), *fn*+1(*x*)>, *x* = 0, 1, 2, … , а *fi* – ПРФ. Покладемо

*F*(*x*1, … , *xn*, *i*) = ⎢*f*1(*i*) – *x*1 ⎢+ … + ⎢*fn*(*i*) – *xn*)⎢, a

*G*(*x*) = *fn*+1(*x*).

Якщо *f* в точці <*x*1, ... , *xn*> визначена, то ця точуа належить *Gf*, a, отже, існує *і* таке, що *F*(*x*1, … , *xn*, *i*) = 0. Значення функції *f* в цьому випадку дорівнює *fn*+1(*i*). Якщо *f* в точці <*x*1, ... , *xn*> не визначена, то алгоритм працює нескінченно довго.

**Зауваження.** Теорему 1 можна сформулювати і так: кожна ЧРФ *f* (*x*1, ... , *xn*> може бути представлена у формі

*f*( *x*1, ... , *xn* ) = *G*(*μi*(*F*(*x*1, … , *xn*, *i*) = 0)).

**Теорема 10.3.** Кожна ЧРФ *f*(*x*) може бути обчислена алгоритмом

function *f*(*x*)

begin

*i* := 0

while *F*(*x*, *i*) ≠ 0 do

*i* := *i* + 1

*f* := ***l***(*i*)

end,

де *F* – деяка (залежна від *f*) ПРФ.

Доведення. Існує ПРФ *g*(*a*, *b*, *z*) така, що рівняння *g*(*a*, *b*, *z*) = 0 має розв’язок ⇔ <*a*, *b*> ∈ *Gf*. Графік *Gf*функції *f* є РПМ. Тому часткова характеристична функція для *Gf* обчислюється наступним алгоритмом:

function χ*G*(*x*, *y*)

begin

*i* := 0

while *g*(*x*, *y*, *i*) ≠ 0 do

*i* := *i* + 1

χ*G* := 0

end,

Покладемо *F*(*x*, *i*) = *g*(*x*, ***l***(*i*), ***r***(*i*)). Тоді функція *f* може бути обчислена наступним алгоритмом:

function *f*(*x*)

begin

*i* := 0

while *F*(*x*, *i*) ≠ 0 do

*i* := *i* + 1

*f* := ***l***(*i*)

end.

Дійсно, нехай функція *f* визначена в точці *x*0. Тоді пара <*x*0, *f*(*x*0) > ∈ *Gf*. Тому рівняння

*g*(*x*0, *f*(*x*0), *z*) = 0

має розвязок *z*0. Нехай ***c***(*f*(*x*0), *z*0) = *i*. Тоді *f*(*x*0) = ***l***(*i*), *z*0 = ***r***(*i*).

Якою б не була пара <*f*(*x*0), z0)> завжди існує *і* – номер цієї пари (***с*** – бієкція).

**Теорема 10.4.** Кожна ЧРФ *f*(*x*) може бути обчислена алгоритмом

function *f*(*x*)

begin

*i* := 0

while *D*(*n*, *x*, *i*) ≠ 0 do

*i* := *i* + 1

*f* := ***l***(*i*)

end,

де *n* – деяке (залежне від *f*) натуральне число.

Доведення. Випливає із теореми 10.1 (*D* – універсальна для класу двохмісних ПРФ).

**Зауваження.** Можна розглянути алгоритм:

function *f*(*x*) function *Т*(*n*, *x*)

begin begin

*f* := *Т*(*n*, *x*) *i* := 0

end while *D*(*n*, *x*, *i*) ≠ 0 do

*i* := *i* + 1

*Т* := ***l***(*i*)

end.

Тоді кожна ЧРФ може бути об числена через функцію *Т*(*n*, *x*) для деякого *n*. Отже, *Т*(*n*, *x*) – універсальна.