КОМПЬЮТЕРНАЯ МАТЕМАТИКА

Вычислительный эксперимент

УДК 681.3

О.І. ПРОВОТАР, О.О. ПРОВОТАР

КОНКРЕТНА АЛГОРИТМІКА

Розглядаються питання конкретизації тези Чорча та її застосування в методології обчислюваності.

Вступ. Всім відома теза Чорча [1, 2] про те, що клас алгоритмічно обчислюваних функцій співпадає з класом частково рекурсивних функцій. Клас частково рекурсивних функцій визначається математично точно. Тому, тезу можна використовувати як для доведення алгоритмічної обчислюваності функцій, так і для доведення того, що функція не є алгоритмічно обчислюваною. Для цього треба лише показати, що така функція належить до класу частково рекурсивних функцій і в цьому випадку вона є алгоритмічно обчислюваною, або не належить і, відповідно, не є алгоритмічно обчислюваною.

Показати алгоритмічну обчислюваність функцій шляхом її належності до класу частково рекурсивних функцій досить складно, окрім найпростіших функцій. Крім того, в кожному конкретному випадку це потребує побудови математичної моделі функції у вигляді терма із обчислюваних операцій над базовими функціями. Базовими функціями, як відомо, [1, 2] називаються найпростіші функції $o(x) = 0$, $s(x) = x + 1$ та функції-селектори $I_m^n(x_1, \dots, x_n) = x_m$, де $n \geq m \geq 1$.

Основними обчислюваними операціями будуть операції суперпозиції $\boldsymbol{S}^{n+1}$, примітивної рекурсії $\boldsymbol{R}$ та мінімізації $\boldsymbol{M}$, які задаються наступними співвідношеннями: операція суперпозиції $\boldsymbol{S}^{n+1}$ дозволяє із n-арної функції $g(x_1, \dots, x_n)$ та n функцій $g_1(x_1, \dots, x_m), \dots, g_n(x_1, \dots, x_m)$, однакової арності утворити функцію

$$f(x_1, \dots, x_m) = g(g_1(x_1, \dots, x_m), \dots, g_n(x_1, \dots, x_m)).$$

Таку функцію позначають $\boldsymbol{S}^{n+1}(g, g_1, \dots, g_n)$.

Операція примітивної рекурсії $\boldsymbol{R}$ дозволяє із n-арної функції g та $n + 2$-арної функції h утворити $n + 1$-арну функцію f за допомогою наступних співвідношень:

$$f(x_1, \dots , x_n, 0) = g(x_1, \dots , x_n),$$

$$f(x_1, \dots , x_n, y + 1) = h(x_1, \dots , x_n, y, f(x_1, \dots , x_n, y)).$$

Таку функцію позначають $\boldsymbol{R}(g, h)$.

Операція мінімізації $\boldsymbol{M}$ дозволяє із n-арної функції g утворити n-арну функцію f, що задається співвідношенням:

$$f(x_1, \dots, x_n) = \mu_y(g(x_1, \dots, x_{n-1}, y) = x_n).$$

Тобто, значення функції $f(x_1, \dots , x_n)$ дорівнює найменшому y для якого $g(x_1, \dots , x_{n-1}, y) = x_n$.

Значення функції $f(x_1, \dots , x_n)$ вважається невизначеним у випадках:

1) для всіх y значення $g(x_1, \dots , x_{n-1}, y) \neq x_n$;

2) для всіх $y < t$ значення $g(x_1, \dots , x_{n-1}, y)$ визначене і $\neq x_n$, а значення $g(x_1, \dots , x_{n-1}, t)$ невизначене;

3) значення $g(x_1, \dots , x_{n-1}, 0)$ невизначене.

Таку функцію позначають $\boldsymbol{M}(g)$.

Конкретизації тези Чорча. Відомо [1, 2], що функцію, яку можна отримати з базових функцій за допомогою скінченної кількості застосувань операцій суперпозиції та примітивної рекурсії, називають примітивно рекурсивною функцією (ПРФ).

Функцію, яку можна отримати з базових функцій за допомогою скінченної кількості застосувань операцій суперпозиції, примітивної рекурсії та мінімізації, називають частково рекурсивною функцією (ЧРФ).

Всюди визначену ЧРФ називають рекурсивною функцією (РФ).

Зрозуміло, що оператори суперпозиції, примітивної рекурсії та мінімізації теж породжують алгоритмічно обчислювані функції із алгоритмічно обчислюваних функцій.

Дійсно, у випадку суперпозиції, якщо ми можемо обчислювати значення функцій g, g_1, … , g_n , то значення їх суперпозиції $f = \boldsymbol{S}^{n+1}(g, g_1, \dots , g_n)$ в точці $(a_1, \dots, a_m)$ можна обчислити за допомогою наступного алгоритму:

```
function f(x1, … , xm)
    begin
        b1 = g1(x1, … , xm)
                    ………………….
          bn = gn(x1, … , xm)
          b  = g (b1, … , bn)
    end.
```

Число b буде значенням функції f в точці $(a_1, \dots , a_m)$.

У випадку примітивної рекурсії, якщо ми можемо обчислювати значення функцій g, h, то значення функції $f = \boldsymbol{R}(g, h)$ в точці $(a_1, \dots , a_n, m + 1)$ може бути обчислене за допомогою наступного алгоритму:

```
function f(x_1, … , x_n, y)
    begin
      if y = 0 then f = g(x_1, … , x_n)
      else f = h(x_1, …, x_n, y – 1, f(x_1, … , x_n, y – 1))
    end.
```

В результаті рекурсивних викликів, отримаємо наступну послідовність чисел:

$$b_0 = g(a_1, \dots , a_n),$$
$$b_1 = h(a_1, \dots , a_n, 0, b_0),$$
$$b_2 = h(a_1, \dots , a_n, 1, b_1),$$
$$\dots\dots ,$$
$$b_{m+1} = h(a_1, \dots , a_n, m, b_m).$$

Число b_{m+1} буде значенням функції f в точці $(a_1, \dots , a_n, m + 1)$.

У випадку операції мінімізації, якщо ми можемо обчислювати значення функції g, то значення функції $f = \boldsymbol{M}(g)$ в точці $(a_1, \dots , a_n)$ може бути обчислене за допомогою наступного алгоритму

```
function f(x_1, … , x_n)
    begin
      i = 0
      while g(x_1, … , x_{n-1}, i) ≠ x_n
          do i = i + 1
      f = i
    end.
```

Іншими словами, ми послідовно обчислюємо значення

$$g(a_1, \dots , a_n, 0),$$
$$g(a_1, \dots , a_n, 1),$$
$$\dots .$$

Найменше значення a, для якого

$$g(a_1, \dots , a_{n-1}, a) = a_n$$

і є значенням функції f в точці $(a_1, \dots , a_n)$.

Виникає цілком закономірне питання про можливість використання конкретних алгоритмів для доведення фундаментальних результатів теорії рекурсивних функцій. Для цього потрібно точно описати основні конструкції алгоритму і переформулювати (конкретизувати) тезу Чорча для більш вузьких класів алгоритмічно обчислюваних функцій.

Домовимося далі записувати алгоритми в мові ПсевдоPaskal, яка є спрощеним діалектом мови Paskal. Операторами цієї мови будуть наступні:

<ідентифікатор> = <вираз>,

if <вираз> then <оператор> | {<оператор>, … , <оператор>}

else <оператор> | {<оператор>, … , <оператор>},

while <вираз> do <оператор> | {<оператор>, … , <оператор>},

for <вираз> to <вираз> <оператор> | {<оператор>, … , <оператор>}.

Конкретизуємо тезу Черча для класів ПРФ, РФ та ЧРФ відповідно.

1. Клас функцій, що обчислюються всюди визначеними алгоритмами без використання оператора while … do. Цей клас описується наступним чином:

1) вважатимемо, що найпростіші функції обчислюються всюди визначеними алгоритмами без використання оператора while … do, а, отже належать до цього класу;

2) всі функції, які обчислюються всюди визначеними алгоритмами без використання оператора while … do і при їх обчисленні використовуються функції, які обчислюються всюди визначеними алгоритмами без використання оператора while … do теж відносимо до цього класу.

Таким чином, справедлива наступна

Теза 1. Клас функцій, що обчислюються всюди визначеними алгоритмами без використання оператора while … do співпадає з класом примітивно рекурсивних функцій.

2. Клас функцій, що обчислюються всюди визначеними алгоритмами. Цей клас описується наступним чином:

1) всі ПРФ належать до цього класу;

2) всі функції, які обчислюються всюди визначеними алгоритмами і при їх обчисленні використовуються функції, які обчислюються всюди визначеними алгоритмами теж відносимо до цього класу.

Таким чином, справедлива наступна

Теза 2. Клас функцій, що обчислюються всюди визначеними алгоритмами співпадає з класом рекурсивних функцій.

3. Клас функцій, що обчислюються довільними алгоритмами. Цей клас описується наступним чином:

1) всі РФ належать до цього класу;

2) всі функції, які обчислюються довільними алгоритмами і при їх обчисленні використовуються функції, які обчислюються довільними алгоритмами теж відносимо до цього класу.

Таким чином, справедлива наступна

Теза 3. Клас функцій, що обчислюються довільними алгоритмами співпадає з класом частково рекурсивних функцій.

Конкретні алгоритми. Покажемо, як доводити алгоритмічну обчислюваність функцій, використовуючи запропоновані вище тези. Нехай, треба довести, що функція

$$f(x,y)=\begin{cases}[x/y], & y\neq 0\\ x, & y=0\end{cases}$$

ПР функція.

Розглянемо послідовність

$$1y \dot{-} x,\ 2y \dot{-} x,\ \dots,\ [x/y] \dot{-} x,\ \dots,\ xy \dot{-} x.$$

Оскільки частка $[x/y]$ означає скільки разів число y «поміщається» в числі x, то $[x/y]$ дорівнює числу нулів у цій послідовності. Дійсно, якщо, наприклад, y два рази «поміщається» в числі x, то

$$1y \dot{-} x = 0,\ 2y \dot{-} x = 0, \text{ а } 3y \dot{-} x \neq 0.$$

Тому алгоритм обчислення функції наступний:

```
function f(x, y)
    begin
      s = 0
      if y = 0 then f = x
      else {for i = 1 to x
             if iy ∸ x = 0 then s = s + 1
              f = s}
    end.
```

Отже, функція $f(x, y)$ є ПРФ.

Якщо всі пари натуральних чисел розташувати в послідовність

$$<0, 0>, <0, 1>, <1, 0>, <0, 2>, <1, 1>, <2, 0>, <0, 3>, \dots ,$$

тобто, впорядкувати всі пари так, що пара $<x, y>$ іде раніше за пару $<u, v>$, якщо

$$x + y < u + v,$$

або якщо

$$x + y = u + v \text{ і } x < u,$$

то співвідношенням

$$<x, y> \leftrightarrow n,$$

де n – номер пари в цій послідовності, задається бієкція між множиною пар та множиною натуральних чисел.

Така бієкція називається нумерацією Кантора пар чисел і позначається $\boldsymbol{c}(x, y)$, тобто $\boldsymbol{c}(x, y)$ – це номер пари $<x, y>$ у послідовності Кантора.

Лівий та правий елементи пари $<x, y>$ з номером n визначають функції $\boldsymbol{l}(n)$ і $\boldsymbol{r}(n)$, які називаються лівою та правою координатними функціями.

Покажемо, що функції $\boldsymbol{c}(x, y)$, $\boldsymbol{l}(n)$, $\boldsymbol{r}(n)$ – ПР функції.

Дійсно, функція $\boldsymbol{c}(x, y)$ обчислюється наступним алгоритмом:

```
function c(x, y)
    begin
      s = 0
      for i = 0 to (x + y)
          s = s + i
      for i = 0 to (x + y)
          {j = (x + y) ∸ i
           if  x = i ∧ y = j then k = i}
       c = s + k
    end.
```

Враховуючи, що $\boldsymbol{l}(n) \le n$, $\boldsymbol{r}(n) \le n$ функція $\boldsymbol{l}(n)$ обчислюється алгоритмом:

```
function l(n)
    begin
      for i = 0 to n
      for j = 0 to n
          if c(i, j) = n  then l = i
    end,
```

а функція $\boldsymbol{r}(n)$ обчислюється алгоритмом:

```
function r(n)
    begin
      for i = 0 to n
      for j = 0 to n
          if c(i, j) = n  then r = j
    end.
```

Таким чином, $\boldsymbol{c}(x, y)$, $\boldsymbol{l}(n)$ та $\boldsymbol{r}(n)$ – ПР функції.

Графіком функції $F(x_1, \ldots, x_n)$ [1] називається сукупність $(n + 1)$-ок вигляду $<x_1, \ldots, x_n, F(x_1, \ldots, x_n)>$.

Покажемо, що якщо графік G_f всюди визначеної функції $f(x_1, \ldots, x_n)$ є рекурсивно перелічимою множиною [ ], то функція f рекурсивна.

Дійсно, графік G_f – це сукупність $(n + 1)$-ок вигляду:

$$<f_1(t), \ldots, f_n(t), g(t)>,$$

де f_i, g – ПРФ. Тоді значення функції f в довільній точці можна обчислити за допомогою наступного алгоритму:

```
function f(x1, … , xn)
  begin
     i = 0
     while f1(i) ≠ x1 ∨ … ∨ fn(i) ≠ xn
          do i = i +1
     f = g(i)
  end,
```

отже, функція f рекурсивна.

Висновки. Таким чином, в статті пропонується підхід, за допомогою якого належність функцій до класів алгоритмічно обчислюваних можна аргументувати побудовою відповідних алгоритмів, на відміну від алгебраїчних термів. Він може бути корисним для програмістів різних вікових категорій, які хочуть познайомитися з основами теорії обчислюваності та теорії алгоритмів, а також студентам відповідних спеціальностей.

А.И. Провотар, А.А. Провотар

КОНКРЕТНАЯ АЛГОРИТМИКА

Рассматриваются вопросы конкретизации тезисов Чорча и их применения в методологии вычислимости.

O.I. Provotar, *O.O. Provotar*

CONCRETE ALGORYTHMICS

The questions of Cherch's theses specification and its application in the methodology of computability are considered.

1. *Мальцев А.И.* Алгоритмы и рекурсивные функции. М.: Наука, 1965. 390 с.
2. *Успенский В.А., Семенов А.Л.* Теория алгоритмов: основные открытия и приложения. М.: Наука, 1987. 288 с.

Одержано 31.08.2016

Про авторів:

Провотар Олександр Іванович,
доктор фізико-математичних наук,
профессор Київського національного університету імені Тараса Шевченка,
professor of the University of Rzeszow (Poland)
E-mail: *aprowata1@bigmir.net*

Провотар Олександр Олександрович,
аспірант Київського національного університету імені Тараса Шевченка.
E-mail: *aprovata@gmail.com*