ძირითადი კონცეფციები რეკურსიული ფუნქციების შესახებ

მთელ რიცხვებზე რეკურსია

როგორც უკვე დავრწმუნდით წინა თავებში, მრავალი ფუნქცია სავსებით ბუნებრივად განისაზღვრება სხვა ფუნქციათა გამოყენებით. მაგალითად, შესაძლებელია განვსაზღვროთ ფუნქცია, რომელიც გვიბრუნებს არაუარყოფითი მთელი რიცხვის ფაქტორიალს საბიბლიოთეკო ფუნქციების გამოყენებით ერთსა და მოცემულ მნიშვნელობას შორის მოთავსებულ რიცხვთა ნამრავლის გამოსათვლელად:

factorial	::	*Int* → *Int*
factorial n	=	*product* [1..*n*]

ჰასკელში დაშვებულია აგრეთვე ფუნქციათა განსაზღვრა საკუთარი თავის გამოყენებით. ასეთ შემთხვევაში ფუნქციებს *რეკურსიულს* უწოდებენ. მაგალითად, ამ გზით შეიძლება განისაზღვროს *factorial* ფუნქციაც:

factorial 0	=	1
factorial (*n* + 1)	=	(*n* + 1)∗ *factorial n*

პირველი განტოლება გვეუბნება, რომ ნულის ფაქტორიალი ერთია და ამ განტოლებას *საბაზო* (*საყრდენი*) *შემთხვევა* (*გამოსახულება*) ეწოდება. მეორე განტოლება ამტკიცებს, რომ ნებისმიერი მკაცრად დადებითი მთელი რიცხვის ფაქტორიალი წარმოადგენს ამ რიცხვისა და მისი წინამავალი რიცხვის ფაქტორიალის ნამრავლს. მას *რეკურსიული შემთხვევა* (*გამოსახულება*) ეწოდება.

მაგალითად, შემდეგი ფრაგმენტი გვიჩვენებს, თუ როგორ ხდება სამის ფაქტორიალის გამოთვლა ამ განსაზღვრების გამოყენებით:

factorial 3
= { *factorial* ფუნქციის გამოყენება }
3 ∗ *factorial* 2
= { *factorial* ფუნქციის გამოყენება }
3 ∗ (2 ∗ *factorial* 1)
= { *factorial* ფუნქციის გამოყენება }
3 ∗ (2 ∗ (1 ∗ *factorial* 0))
= { *factorial* ფუნქციის გამოყენება }
3 ∗ (2 ∗ (1 ∗ 1))
= { ∗ ოპერაციის გამოყენება}
6

ყურადღება მიაქციეთ იმ გარემოებას, რომ, თუმცა *factorial* ფუნქცია საკუთარი თავის საშუალებით განისაზღვრება, იგი უსასრულო ციკლს არ წარმოადგენს. სახელდობრ, *factorial* ფუნქციის ყოველი გამოყენება ამცირებს მთელრიცხვა არგუმენტს ერთით, ვიდრე იგი საბოლოო ჯამში ნულის ტოლი არ გახდება. აქ რეკურსია წყდება და გამრავლების ოპერაცია სრულდება. ნულის ფაქტორიალი, რომლითაც ერთიანი გვიბრუნდება, სავსებით ადეკვატურია ამ ვითარებაში, რადგან გამრავლებისას ერთიანი იგივეობას უნარჩუნებს გამოსახულებას. სხვანაირად რომ ვთქვათ, $1 * x = x$ და $x * 1 = x$ ნებისმიერი მთელი x რიცხვისათვის.

factorial ფუნქციის შემთხვევაში საწყისი განსაზღვრება საბიბლიოთეკო ფუნქციების საშუალებით უფრო მარტივია, ვიდრე რეკურსიის გამოყენებით. მაგრამ, როგორც ამას წიგნის დარჩენილ ნაწილში დავინახავთ, მრავალი ფუნქცია მარტივად და ბუნებრივად სწორედ რეკურსიის გამოყენებით განისაზღვრება. მაგალითად, მრავალი საბიბლიოთეკო ფუნქცია ჰასკელში განისაზღვრება რეკურსიით. გარდა ამისა, როგორც მე-13 თავში ვნახავთ, ფუნქციათა განსაზღვრა რეკურსიით მათი თვისებების დამტკიცების საშუალებას იძლევა მათემატიკური ინდუქციის მძლავრი მეთოდის გამოყენებით.

მთელ რიცხვებზე რეკურსიის კიდევ ერთ მაგალითად განვიხილოთ ზემოთ გამოყენებული გამრავლების * ოპერატორი. ეფექტურობის მოსაზრებებიდან გამომდინარე, ჰასკელში ეს ოპერატორი გათვალისწინებულია როგორც პრიმიტივი – დაპროგრამების ენაში პროგრამათა შესრულების სიჩქარის გაზრდისათვის ჩაშენებული ოპერატორი. მაგრამ არაუარყოფითი მთელი რიცხვებისათვის იგი შეიძლება იყოს განსაზღვრული ასევე რეკურსიითაც თავისი ორი არგუმენტიდან ნებისმიერ ერთზე, ვთქვათ მეორეზე:

$$
\begin{array}{lll}
(*) & :: & Int \to Int \to Int \\
m * 0 & = & 0 \\
m * (n + 1) & = & m + (m * n)
\end{array}
$$

მაგალითად:

$4 * 3$

= { * ოპრატორის გამოყენება }

$4 + (4 * 2)$

= { * ოპრატორის გამოყენება }

$4 + (4 + (4 * 1))$

= { * ოპრატორის გამოყენება }

$4 + (4 + (4 + (4 * 0)))$

= { * ოპრატორის გამოყენება }

$4 + (4 + (4 + 0))$

= { + ოპრატორის გამოყენება }

12

ამრიგად, აქ რეკურსიული განსაზღვრება * ოპერატორისათვის ფორმალურად გამოხატავს იმ იდეას, რომ გამრავლება დაიყვანება იტერაციულ (მრავალჯერად) შეკრებამდე.