THIS PAGE IS INTENTIONALLY LEFT BLANK.

Kazimir Majorinc

Temelj za matematičku teoriju izračunavanja

Povijest Lispa 18.

A

Razmjena vještina
Hacklab u mami
12. siječnja 2013.

A basis for a mathematical theory of computation, preliminary report

IRE-AIEE-ACM '61 (Western) Papers presented at the May 9-11, 1961, western joint

Programs that learn to modify their own behaviors require a way of representing algorithms so that interesting properties and interesting transformations of algorithms are simply represented.

Theories of computability have been based on Turing machines, recursive functions of integers and computer programs.

Each of these has artificialities which make it difficult to manipulate algorithms or to prove things about them.

The present paper presents a formalism based on conditional forms and recursive functions whereby the functions computable in terms of certain base functions can be simply expressed.

We also describe some of the formal properties of conditional forms, and a method called recursion induction for proving facts about algorithms.

A basis for a mathematical theory of computation, MIT AIM-031, January 1962.

A basis for a mathematical theory of computation, Computer Programming and Formal Systems (**1963**), pp. 33-70

„This paper is a corrected version of the paper of the same title given at the Western Joint Computer Conference, May 1961. A tenth section discussing the relations between mathematical logic and computation has been added.“

„Članak je pokušaj stvaranja temelja za matematičku teoriju izračunavanja. Prije nego što kažemo što je u članku, kratko ćemo prodiskutirati kojim se praktičnim rezultatima možemo nadati od odgovarajuće matematičke teorije. Ovaj članak sadrži direktne priloge samo nekolicini spomenutih ciljeva, ali popisujeme dodatne ciljeve e da bi ohrabrili potragu za zlatom.

1. Razvoj univerzalnog programskog jezika. Vjerujemo da je znatan broj ljudi prerano otpisao taj cilj. Mislimo da je ALGOL na pravom putu, ali mu nedostaje mogućnost opisa različitih tipova podataka; COBOL je korak na slijepom putu u orijentaciji prema engleskom jeziku koji nije pogodan za formalne opise postupaka, a da je UNCOL vježba u grupnom priželjkivanju („wishfull thinking“).

2. Definirati teoriju ekvivalencije procesa izračunavanja. Takva teorija nam može pomoći da od algoritma koji rješava problem dođemo do ekvivalentnog ali bržeg, manje zahtjevnog i sl. algoritma.

3. Prestaviti algoritme simboličkim izrazima tako da su značajne promjene u ponašanju algoritma malim promjenama u simboličkim izrazima.

4. Predstaviti računala kao i računalne procese.

5. Dati kvantitativnu teoriju izračunavanja.

Formalizmi za opisivanje izračunljivih funkcija i povezanih bića.

1. Funkcije izračunljive u terminima funkcija baze.

Neka je F skup osnovnih parcijalno definiranih funkcija. McCarthyjev cilj je definirati C{F}, klasu funkcija izračunljivih korištenjem F („in terms of F“).

Metode za definiciju funkcija su iste kao pri definiranju S-funkcija. Kompozicija funkcija; iskazni veznici, uvjetni izrazi, rekurzija.

Uvjetni izraz

$$(p1 \rightarrow e1, ..., pn \rightarrow en)$$

ne može se definirati kao funkcija

$$condn(p1, ..., pn, e1, ..., en)$$

jer svi argumenti funkcije moraju biti definirani.

Što bi se dogodilo da dozvolimo funkcije na nedefiniranim argumentima? Ne bi mogli funkcije više varijabli izjednačavati sa funkcijom jedne varijable na produktu prostora.

Npr, ako bi dozvolili da je funkcija dvije varijable, $f(x,y)$, $x\in A$, $y\in B$ definirana i kad x nije definiran, onda ne bi mogli identificirati funkciju dvije varijable sa funkcijom definiranom na produktu $A\times B$, jer bi to tražilo da je i par (nedefiniran,B) u tom produktu.

A zašto treba dozvoliti da neki argumenti uvjetnog izraza budu nedefinirani? Primjer:

$$n! = (n = 0 \rightarrow 1, \mathrm{T}\rightarrow n\cdot(n - 1)!)$$

Izračunavanje 0! bi tražilo da se izračuna (n-1)!

C{F} se sastoji od funkcija koje se mogu definirati ponovljenom primjenom opisanih metoda definiranja.

2. Rekurzivne funkcije nad prirodnim brojevima.

I={0,1,2,...}

succ(0)=1,succ(1)=2,... označava se i 0'=1, 1'=2, ...

eq(0,0)=T,eq(1,1)=T,... označava se i 0=0, 1=1, itd.

F={succ,eq}

pred(n) = pred2(n,0), označava se i n-

pred2(n,m)=(m'=n → m, T →pred2(n,m'))

m+n = (n=0 → m, T → m'+n-)

mn = (n=0 → 0, T →m+mn-)

m-n = (n=0 → m, T → m- - n-)

$m \leq n = (m=0) \vee (\sim (n = 0) \wedge (m- \leq n-))$

$m/n = (m < n \rightarrow 0, T \rightarrow ((m-n)n)')$

$rem(m/n) = (m < n \rightarrow m, T \rightarrow rem((m-n)/n))$

...

C{F} = sve uobičajene aritmetičke funkcije.