Kybernetika

Ľudovít Molnár

Analytické derivovanie na číslicovom počítači

Kybernetika, Vol. 6 (1970), No. 4, (291)--295

Persistent URL: http://dml.cz/dmlcz/125315

KYBERNETIKA ČÍSLO 4, ROČNÍK 6/1970

Analytické derivovanie na číslicovom počítači

Ľudovít Molnár

Článok popisuje algoritmus analytického derivovania výrazov, ktorý je modifikáciou algoritmu navrhnutého Hansenom [2]. Derivovanie sa uskutočňuje v niekoľkých „cykloch", pričom v každom „cykle" sa uskutoční jeden krok derivovania. Úlohou modifikovaného algoritmu bolo znížiť počet „cyklov" na minimum.

V práci sa podáva algoritmus analytického derivovania výrazov a jeho realizovanie na samočinnom číslicovom počítači. Algoritmus vychádza z prepisu výrazu zapísaného v jazyku ALGOL 60 do bezzátvorkového tvaru, na ktorý sa aplikujú známe pravidlá z diferenciálneho počtu. Rozoberá sa iba prvá derivácia, avšak veľmi jednoduchým spôsobom je možné uskutočniť deriváciu ľubovolného rádu.

Počiatočná informácia pre program je prípustný výraz zapísaný v jazyku ALGOL a premenná derivovania.

ALGORITMUS

Podstata algoritmu spočíva v aplikácii vhodných transformačných vzorcov na výraz zapísaný v bezzátvorkovom tvare. Pri tvorbe algoritmu budeme vychádzať z práce I. W. Hansena a kol. [2]. Jeho transformačné vzorce môžeme písať v tvare:

1 $$[\pm uv]' = \pm [u]'\,[v]',$$
2 $$[\div u]' = \div [u]',$$
3 $$[\times uv]' = + \times [u]'\,v \times u[v]',$$
4 $$[\uparrow nu]' = \times [u]' \times \uparrow -1nun,$$
5 $$[\exp u]' = \times [u]' \exp u,$$
6 $$[\ln u]' = \times \uparrow \div 1u[u]',$$
7 $$[\sin u]' = \times [u]' \cos u,$$
8 $$[\cos u]' = \times [u]' \div \sin u,$$
9 $$[z]' = 1,$$
10 $$[\text{konšt.}]' = 0,$$

kde ′ znamená znak derivovania,

$\div$ je operátor zmeny znamienka,
u, v sú ľubovoľné prípustné výrazy,
n je ľubovoľný prípustný výraz, ktorý neobsahuje premennú derivovania,
z je premenná derivovania.

Pod prípustným výrazom budeme rozumieť v podstate jednoduchý aritmetický výraz definovaný v jazyku ALGOL, rozšírený o operátor zmeny znamienka a obsahujúci iba tie funkcie, ktoré sú obsiahnuté v transformačných vzorcoch. Transformačné vzorce môžeme pochopiteľne vhodne rozšíriť.

Celý proces derivácie prebieha v niekoľkých „cykloch", ktorých počet závisí od tvaru výrazu, ktorý máme derivovať. V jednom „cykle" sa uskutoční iba jeden krok derivovania, napr. derivácia vonkajšej funkcie, derivácia súčinu bez derivácie činiteľov ap.

Určité zjednodušenie uvedeného algoritmu navrhol R. G. Tobey [5]. Zjednodušenie sa robí tým, že sa pred použitím transformačných vzorcov urobí analýza príslušného výrazu, ktorý chceme derivovať, čím sa v mnohých prípadoch transformačné vzorce zjednodušia, ako napr. pri derivácii súčinu ak jeden operand je konštanta a pod.

Napriek uvedenému zjednodušeniu musíme pri tomto algoritme určitým spôsobom pracovať i s prvkami výrazu, ktoré tvoria výsledok niektorého „cyklu" derivovania a nemenia a ani nebudú meniť svoj tvar. Tento problém by sme odstránili, keby sme celý proces zlinearizovali, tj. celý proces derivovania by sa uskutočnil v jednom „cykle" derivovania. V ďalšom si ukážeme algoritmus, ktorý sme vypracovali na Katedre matematických strojov EF SVŠT. Tento algoritmus síce neprevedie celý proces derivovania vždy na lineárny, ale vo väčšine prípadov podstatne obmedzí počet „cyklov" derivovania.

Transformačné vzorce pri našom algoritme nadobudnú tvar:

1	$\pm = \pm$
2	$\div = \div$
3	$\times = + \times [u\ Ip\ v \times u\ v]'$
4	$\uparrow = \times + \times \uparrow - 1\ n\ u\ n$
5	$\exp = \times \exp u$
6	$\ln = \times \uparrow \div 1\ u$
7	$\cos = \times \div \sin u$
8	$\sin = \times \cos u$
9	$z = 1$
10	konšt. $= 0$

Prv než uvedieme popis algoritmu, rozoberieme si podrobnejšie transformačné vzorce pre násobenie a umocňovanie. Celý proces derivovania podľa nášho algoritmu bude prebiehať tak, že sa v podstate derivuje každý prvok výrazu. K tomu sú

vhodne prispôsobené i naše transformačné vzorce. Zoberme si však transformačný vzorec pre násobenie z pôvodného algoritmu

$$[\times\ uv]' = +\ \times\ [u]'\ v\ \times\ u\ [v]'\ .$$

Medzi výrazmi na pravej strane, ktoré potrebujeme derivovať, tj. $[u]', [v]'$, sa nachádzajú výrazy u, v a operátor $\times$, na ktoré sa nebudú aplikovať transformačné vzorce, ale ktoré iba opíšeme. Preto náš nový transformačný vzorec, ktorý vychádza z toho, že budeme derivovať každý prvok, obsahuje dvojicu prvkov $I\ p$, kde p je počet prvkov z ktorých sa skladajú výrazy u, v zväčšený o jednotku (operátor $\times$), ktorá nám určuje, že nasledujúcich p prvkov derivovať nebudeme, ale ich iba opíšeme.

Z rovnakých dôvodov obsahuje i transformačný vzorec pre umocňovanie operátor $+$. Ak totiž budeme derivovať výraz tvoriaci exponent (neobsahuje premennú derivovania), dostaneme ako výsledok nulu, ktorá bude viazaná uvedeným operátorom $+$.

Ako sme už uviedli, celý proces derivovania podľa nášho algoritmu prebieha tak, že sa derivuje každý prvok výrazu. Už priamo z transformačných vzorcov vidieť, že v prípade, že výraz neobsahuje operátor $\times$, celý proces derivovania je lineárny. V prípade, že výraz obsahuje operátor $\times$, zväčší sa počet „cyklov" derivovania o toľko, koľko je operátorov $\times$. Celý proces prebieha nasledujúcim spôsobom:

1. prepis výrazu do bezzátvorkového tvaru,
2. priradenie znaku derivácie a koncového znaku,
3. vyhľadanie znaku derivácie,
4. analýza nasledujúceho prvku.

Poradie ďalších krokov závisí od druhu prvku, ktorý sme analyzovali. Ak je to koncový znak, pokračujeme krokom 7., ak je to operátor $\times$, aplikuje sa príslušný transformačný vzorec a zbývajúce prvky sa prepíšu; pokračuje sa krokom 3., ak je to znak I zoberie sa nasledujúci prvok p, ktorý nám určí koľko ďalších prvkov sa iba prepíše a pokračuje sa krokom 4., inak sa pokračuje krokom 5.

5. Vyhľadanie príslušného transformačného vzorca a jeho aplikácia,
6. skok na krok 4,
7. výstup.

Krokom 7. sme dostali zderivovaný výraz v bezzátvorkovom tvare. Ďalej môže nasledovať jeho použitie v uvedenom tvare, alebo prepis do ALGOLu.

URČENIE PODVÝRAZOV

Aby sme transformačné vzorce vôbec mohli použiť, musíme si najprv určiť podvýrazy, ktoré tvoria operandy jednotlivých operácií. Pre určenie podvýrazu bude platiť pravidlo: podvýraz bude tvorený postupnosťou (reťazcom) prvkov z daného

výrazu, v ktorom počet operandov je o jednotku väčší, ako počet binárnych operátorov $(+, -, \times, \uparrow)$.

Napr. vo výraze

$$+, \uparrow, 2, \sin, \times, 3, z, \uparrow, 2, z,$$

ktorý je tvaru $+uv$, je podvýraz u tvorený reťazcom

$$\uparrow, 2, \sin, \times, 3, z$$

a podvýraz v reťazcom

$$\uparrow, 2, z$$

ako vidieť, oba reťazce vyhovujú nášmu pravidlu.

POUŽITIE

Použitie oboch algoritmov si ukážeme na jednoduchom príklade. Majme derivovať výraz:

$$z^2 + \sin^2 3z .$$

Prepisom do bezzátvorkového tvaru a priradením znaku derivácie dostaneme:

$$[+, \uparrow, 2, \sin, \times, z, 3, \uparrow, 2, z]' .$$

Derivácia pôvodným algoritmom bude mať tvar:

1. $+, [\uparrow, 2, \sin, \times, z, 3]', [\uparrow, 2, z]'$
2. $+, \times, [\sin, \times, z, 3]' \times, \uparrow, -, 1, 2, \sin, \times, z, 3, 2, \times, [z]', \times, \uparrow, -, 1, 2, z, 2$
3. $+, \times, \times, [\times, z, 3,]', \cos, \times, z, 3, \times, \uparrow, -, 1, 2, \sin, \times, z, 3, 2, \times, 1, \times, \uparrow, -, 1, 2, z, 2$
4. $+, \times, \times, +, \times, [z,]', 3, \times, z, [3,]', \cos, \times, z, 3, \times, \uparrow, -, 1, 2, \sin, \times, z, 3, 2, \times, 1, \times, \uparrow, -, 1, 2, z, 2$
5. $+, \times, \times, +, \times, 1, 3, \times, z, 0, \cos, \times, z, 3, \times, \uparrow, -, 1, 2, \sin, \times, z, 3, 2, \times, 1, \times, \uparrow, -, 1, 2, z, 2$

Výsledkom je zderivovaný výraz zapísaný v bezzátvorkovom tvare. Prepisom do ALGOLu dostaneme:

$$2 \times z \uparrow (2 - 1) \times 1 + 2 \times \sin (3 \times z) \uparrow (2 - 1) \times \cos (3 \times z) \times (0 \times z + 3 \times 1)$$

Výraz možno podstatne zjednodušiť vykonaním elementárnych operácií.

Derivácia novým algoritmom bude mať tento tvar:

1. $+, \times, +, \times, \uparrow, -, 1, 2, \sin, \times, z, 3, 2, 0, \times, \cos, \times, z, 3, +, \times, [z, I, 3, 3, \times, z, 3, \uparrow, 2, z]'$
2. $+, \times, +, \times, \uparrow, -, 1, 2, \sin, \times, z, 3, 2, 0, \times, \cos, \times, z, 3, +, \times, 1, 3, \times, z, 0, \times, +, \times, \uparrow, -, 1, 2, z, 2, 0, 1$

Prepísaním do ALGOLu dostaneme rovnaký výsledok ako v predošlom prípade. Ako vidieť z príkladu, počet „cyklov" derivovania sa skutočne podstatne zmenšil.

PROGRAM

Pri programovaní treba brať do úvahy to, že veličiny, ktoré budeme spracovávať nemusia mať číselný charakter. Preto ak chceme s takýmito veličinami pracovať napr. v jazyku ALGOL je potrebné urobiť najprv vhodné prekódovanie. Z tohoto dôvodu nie je jazyk ALGOL vhodný na programovanie úloh podobného charakteru. Najvýhodnejšie je použiť strojový, alebo vhodný symbolický kód, resp. programovací jazyk umožňujúci spracovávať obecné symboly.

Program bol pôvodne napísané v jazyku ALGOL a neskôr prepísaný do strojového kódu počítača MINSK 22 na ktorom bol celý algoritmus overovaný.

(Došlo dňa 14. októbra 1968.)

LITERATÚRA

[1] J. W. Backus a kol.: Programovaní v jazyku ALGOL 60. SNTL, Praha 1963.
[2] I. W. Hanson: Analytic differentiation by computer. Comm. ACM *5* (June 1962), 6, 349—355.
[3] H. G. Kahrimanian: Analytical Differentiation on a Digital Computer. Temple University, May 1953.
[4] B. Randel, L. J. Russel: ALGOL 60 Implementation. Pergamon Press, London 1964.
[5] R. C. Tobey, R. J. Bobrow, S. Zilles: Experience with FORMAC Algorithm Design. Comm. ACM *9* (Aug. 1966), 8, 589—597.

SUMMARY

Analytical Differentiation on a Digital Computer

ĽUDOVÍT MOLNÁR

The paper describes an algorithm for analytical differentiation of the expressions which is a modification of an algorithm suggested by Hansen [2]. Differentiation is done in a few "loops" and in every "loop" one step of differentiation is done. The task of the modified algorithm is to reduce the number of "loops" to the minimum.

Ľudovít Molnár, prom. fyz., Katedra matematických strojov, EF SVŠT, Vazovova 1/b, Bratislava.