

Ladislav Koubek

Zobrazení veličin a specifických parametrů procedur v jedné realizaci překladače z jazyka Algol 60

Acta Universitatis Carolinae. Mathematica et Physica, Vol. 10 (1969), No. 1-2, 71--76

Persistent URL: <http://dml.cz/dmlcz/142234>

Terms of use:

© Univerzita Karlova v Praze, 1969

Institute of Mathematics of the Academy of Sciences of the Czech Republic provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This paper has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ: The Czech Digital Mathematics Library* <http://project.dml.cz>

ZOBRAZENÍ VELIČIN A SPECIFIKACE PARAMETRŮ PROCEDUR
V JEDNÉ REALIZACI PŘEKLADAČE Z JAZYKA ALGOL 60

L. KOUBEK

Centrum numerické matematiky UK, Praha

ИЗОБРАЖЕНИЕ ВЕЛИЧИН И СПЕЦИФИКАЦИЯ ПАРАМЕТРОВ ПРОЦЕДУР В ОДНОМ ТРАНСЛЯТОРЕ С ЯЗЫКА АЛГОЛ 60. В работе описывается способ, с помощью которого транслятор с языка АЛГОЛ 60 работает с величинами целочисленными, вещественными и логическими.

Все три типа величин представляются с плавающей запятой, благодаря чему возможно легкое программирование выражений, в которых встречаются величины разных типов, и опущение спецификации параметров процедур.

В операторах присваивания приходится, однако, зачислять программу функции перевода или соотв. модифицированную функцию перевода. Оба эти понятия в работе определены.

ALGOL 60 povoluje používat veličiny typu real i integer v aritmetickém výraze bez jakéhokoli omezení. Rozdíl mezi oběma typy je jen v aritmetice, tj. s celými čísly je nutno provádět operace přesně, u operací s čísly reálnými se předpokládá, že bude provedeno zaokrouhlování (závislé na použitém počítači a v ALGOLu 60 blíže nespecifikované), takže dojde k jisté nepřesnosti výsledku.

ALGOL 60 je určen především k zápisu algoritmů vědecko-technických výpočtů. V těchto výpočtech používáme celá čísla většinou jako indexy. Z tohoto důvodu je nejjednodušší volit pro oba typy stejné zobrazení a to v pohyblivé řádové čárce.

Výhody této volby zobrazení čísel jsou zřejmé. Nenastanou žádné

komplikace při programování výrazů, v nichž používáme veličiny různých typů.

Na druhé straně zmenšujeme přípustný interval celých čísel při každé realizaci překladače na počítači, neboť nejen celá čísla, ale také všechny mezivýsledky, které při práci s nimi vznikají, musejí být zobrazitelné v části slova vyhrazené pro mantisu. Jinak nebude výsledek typu integer, neboť se provede zaokrouhlení i když teoreticky zůstaneme v okruhu celých čísel.

U většiny počítačů dojde také k jistému zpomalení výpočtů, protože operace v pevné řádové čárce obvykle pracují rychleji než operace v pohyblivé řádové čárce. Protože však odpadají převody z jednoho zobrazení do druhého, nemusí být tato ztráta, v programech psaných v ALGOLu 60, podstatná. V realizaci PHEN-ALGOL /1/ byla tato volba dokonce optimální, neboť na počítači ODRA 1003 jsou operace v pevné i pohyblivé řádové čárce stejně rychlé (s výjimkou dělení).

Pokud bychom byli nuceni volit různá zobrazení pro oba typy čísel, můžeme z výhodou využít toho, že algoritmus překladače je stavěn tak, abychom podle typu prvního symbolu mohli měnit operační tabulku a ihned při načítání dalších symbolů zařazovat do programu příslušné transformační funkce. Hodnoty transformovaných proměnných lze umístit do zvláštních pracovních buněk a v dalším pracovat už jen s adresami těchto pomocných buněk. Podrobněji nebudeme v této práci o tomto problému mluvit, ale vrátíme se k případu, kdy oba typy čísel mají stejná zobrazení.

Jsou-li zobrazení obou typů čísel v počítači stejná, je nutno podle syntaxe ALGOLu 60 provádět transformaci jen v tom případě, že

- a) jedna (a tedy všechny) levá část dosazovacího příkazu je typu integer a výraz E na pravé straně má hodnotu typu real a
- b) při výpočtu hodnoty indexových výrazů typu real. Transformační funkce má v obou případech tvar

entier (E + 0.5).

Tuto transformační funkci realizujeme na většině počítačů velmi snadno několika instrukcemi, protože jde jen o zaokrouhlení podle obvyklých pravidel a odříznutí zlomkové části.

Na počítači ODRA 1003 jsou to 3 instrukce.

307 17021 + 00

277 00000 + 70

(1)

177 00000 + 70

V buňce 17021 je uložena konstanta 0.5 v nenormalizovaném tvaru $2^{-31} \cdot 2^{30}$. Nenormalizovaným přičtením této hodnoty k normalizované hodnotě ve střadači posuneme obraz výsledku tak, že zlomková část je umístěna do 32. až 38. bitu. Druhá instrukce, kterou nenormalizovaný obsah střadače přesíláme do rychlé buňky paměti, odřízne tuto zlomkovou část a nahradí ji řádem. Třetí instrukcí pak jen vrátíme výsledek do střadače, ale normalizovaný.

Z čistě technických důvodů nebylo možno použít jedinou instrukci normalizovaného sčítání, která by provedla zaokrouhlení, odříznutí zlomkové části (vysunutím ze střadače) a normalizaci, neboť počítače ODRA 1003 i 1013 jsou schopny při sčítání vyrovnávat jen řády, jejichž rozdíl je menší než 37, což by vedlo k chybným výsledkům.

Aby bylo možno zařadit program transformační funkce, je nutno poněkud pozměnit pravidla C 2 a B uvedená v /2/.

Především je třeba do operační tabulky přidat ke každému operátoru příznak určující, zda jde o operátor z okruhu celých čísel nebo nikoliv. Volíme například příznak 0 pro operátory okruhu celých čísel a 1 pro operátory ostatní.

Podobně v tabulkách proměnných, polí, funkcí atd. označujeme identifikátory typu integer příznakem 0, identifikátory typu real příznakem 1 a nespécifikované formální parametry procedur příznakem 2.

Při práci podle pravidel A 0.1. a A 0.2. přenášíme tyto příznaky do pracovní buňky překladače Q, ale jen v tom případě, že nový údaj je větší než dosavadní obsah Q. Pravidlo C 2 změním na:

Pravidlo C 2': Je-li načten oddělovač $:=$, zapíšeme do mikroprogramu M instrukci H<A>, obsah buňky Q dáme do buňky Q 1. Do A zapíšeme adresu nuly, do S dosadíme instrukci B, Q vynulujeme a pokračujeme podle A 0.

Pravidlo B bude mít tento tvar:

Pravidlo B 2: Do programu zařazujeme mikroprogramy sestupně podle indexu i. Je-li $i \geq 1$ a první buňka

mikroprogramu s indexem $i-1$ je nenulová, zařadíme instrukci Hp_i .

Zařazování mikroprogramů ukončíme po zapsání mikroprogramu s indexem 0.

Je-li alespoň jedna buňka mikroprogramu M nenulová, pak když

- a) $\langle Q \rangle = 0$, $\langle Q \ 1 \rangle = 1$ nebo 2, zařadíme program transformační funkce (1).
- b) $\langle Q \rangle = 2$, $\langle Q \ 1 \rangle = 1$ nebo 2 zařadíme modifikovaný program transformační funkce (2).
- c) $\langle Q \rangle = 1$, nebo $\langle Q \ 1 \rangle = 0$ nezařazujeme program transformační funkce.

Ve všech třech případech přepíšeme nenulové instrukce mikroprogramu M a pole mikroprogramu vymažeme. Pak pokračujeme podle jednoho z pravidel $D(t)$.

Je-li $\langle Q \rangle = 0$, je levá strana dosazovacího příkazu typu integer. Je-li přitom $\langle Q \ 1 \rangle = 1$, je výraz na pravé straně typu real, neboť se v něm vyskytl alespoň jeden operátor, jehož výsledek není celé číslo nebo veličina typu real, takže musíme zařadit program transformace.

Je-li $\langle Q \rangle = 1$, je typ levé strany real a transformaci nemusíme provádět v žádném případě.

Je-li $\langle Q \ 1 \rangle = 0$, je typ výrazu na pravé straně integer, neboť se v něm vyskytují jen celočíselné veličiny a operátory okruhu celých čísel, takže transformaci před dosazením do levé části opět není nutno provádět (ať je typ levé části jakýkoliv).

Je-li konečně $\langle Q \ 1 \rangle = 2$, neznáme v okamžiku kompilace tohoto příkazu typ výrazu na pravé straně, neboť v něm vystupuje jako operand alespoň jeden nespecifikovaný formální parametr, jehož typ bude určen až při vyvolávání této funkční procedury (a to podle typu příslušného skutečného parametru, kterým bude formální parametr nahrazen).

Je-li při tom $\langle Q \rangle = 0$, je levá část dosazovacího příkazu typu integer a program transformace musíme zařadit. Tato transformace může být ovšem také zbytečná, je-li typ skutečného parametru integer, nezpůsobí však chybu ve výpočtu, neboť nijak nezmění

hodnotu výrazu.

Je-li $\langle Q \rangle = 2$, neznáme ani typ levé části dosazovacího příkazu. Bude opět určen typem skutečného parametru při vyvolání procedury. Je-li $\langle Q \rangle \neq 0$, musíme proto zařadit modifikovaný program transformační funkce, tj. takový program, který transformační funkci provede, je-li skutečný parametr typu integer a vynechá, je-li typu real.

Modifikovaný program má stejný tvar jako program transformační funkce, před ním je však instrukce nepodmíněného předání řízení modifikovaná určitým příznakem (hodnotou některého indexového registru), který bude určen až při výpočtu podle přeloženého programu a to v okamžiku, kdy formální parametr nahrazujeme parametrem skutečným a známe tedy již typ levé části prováděného dosazovacího příkazu. Podle tohoto modifikačního příznaku se pak buď transformace provede nebo vynechá. V realizaci PHEN-ALGOL pro počítač ODRA 1003 má modifikovaný program tvar

```
000 00000 + 61
307 17021 + 00
277 00000 + 70
177 00000 + 70
```

(2)

Do šestého registru (který na počítači ODRA modifikuje skokovou adresu) zapisujeme 3 pro skutečné parametry typu real a 0 pro skutečné parametry typu integer.

Zvolený způsob zobrazení čísel obou typů a zavedení modifikovaného programu transformační funkce v podstatě umožnil, že v našem algoritmu (a ovšem i v realizaci PHEN-ALGOL) nemusíme vyžadovat specifikaci parametrů. U parametrů, které používáme buď jako celá nebo reálná čísla to plyne z toho, co jsme právě řekli. Logické hodnoty zobrazujeme jako čísla + 1 a - 1 (typu integer) a jejich specifikaci není proto nutno vyžadovat. Výraz, v kterém smíme pracovat s logickými hodnotami, je v syntakticky správném textu vždy boolovský a nabude rovněž hodnoty + 1 nebo - 1, z hlediska našeho zobrazení je typu integer. Charakter ostatních formálních parametrů, které jsou užity jako identifikátory polí, procedur apod., je určen jejich prvním použitím v operační části procedury, jak o tom bude řeč v práci /3/.

Literatura

- /1/ Koubek L. - Programující program PHEN-ALGOL pro počítače ODRA 1003 a 1013
Závěrečná zpráva č. 10/69 VÚZORT - Praha
- /2/ Koubek L. - Algoritmus kompilace nepodmíněných výrazů, dosazovacích příkazů a příkazů skoku v translátoru z jazyka ALGOL 60, AUC, Math.-Phys., Vol. 10, 57-69 (1969)
- /3/ Koubek L. - Kompilace popisů a příkazů procedur v překladači z jazyka ALGOL 60, AUC, Math.-Phys., Vol. 10, 97-102 (1969)