

Kreslení vývojových diagramů počítačem II

Ing. Václav SKÁLA — Hynek HLAVNÍČKA, katedra technické kybernetiky VŠSE, Plzeň

V článku je popsán algoritmus kreslení vývojových diagramů programu napsaného v jazyce ALGOL 60 pomocí počítače na kreslicím stole DIGIGRAF 1008. Program byl realizován na počítači ODRA 1204 v jazyce ALGOL 60 a je ověřen několikaměsíčním používáním.

Úvod

V souladu s požadavky na technickou dokumentaci ve výpočetní technice je žádoucí, aby počítač kreslil vývojové diagramy přímo ze zdrojového programu. Proto byl sestaven algoritmus umožňující kreslení vývojových diagramů programů napsaných v jazyce ALGOL 60. Značky používané programem jsou převzaty z normy ČSN 36 9030 [1] kromě značky začátku a konce cyklu.

Lexikální a syntaktická analýza

Pro kreslení vývojových diagramů jsme upravili gramatiku jazyka ALGOL 60. Syntaktická analýza je prováděna metodou syntaktických procedur. Výstupem je poněkud složitější datová struktura, která obsahuje úplnou informaci o struktuře programu. V této části se zároveň provádí převod z obecně hnízdové blokové struktury, která je typická pro jazyk ALGOL 60, do lineární struktury, typické pro jazyk FORTRAN. Ošetření syntaktických chyb programu se neprovádí, neboť se předpokládá kreslení odladěných programů.

Generování tabulek pro kreslení

V této části se data získaná syntaktickou analýzou zpracovávají do tabulek, které slouží ke kresle-

ní vývojového diagramu. Algoritmus umísťuje značky do libovolně dlouhého rastru, a ten se pak dělí na stránky. Výstupní data již obsahují informace, potřebné k nakreslení zvolených stránek vývojového diagramu.

Vývojové diagramy kreslí i algoritmus, který je popsán v [5] a který byl doplněn o automatické kreslení šipek.

Značky se rozmísťují počínaje prostředním sloupcem. Pokud se v programu vyskytuje větvení, umísťují se větve střídaně vpravo a vlevo od prostředního sloupce. Toto řešení je esteticky výhodné. Rozmísťování probíhá rekurzivním způsobem.

Stránkování se provádí především proto, aby bylo možno přikládat nakreslené vývojové diagramy k celkové dokumentaci programu.

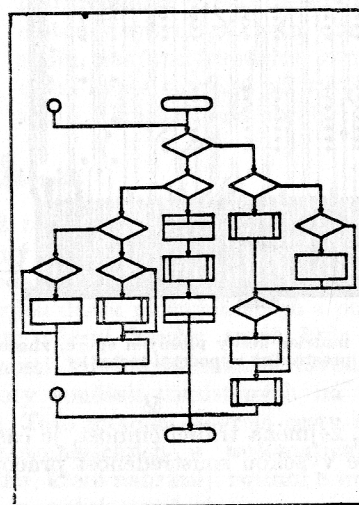
Závěr

Realizovaný program podstatně usnadňuje dokumentační proces ve výpočetové technice, která nebývá obvykle důsledná. Lze konstatovat, že uvedený algoritmus lze použít pro dokumentaci programů napsaných v jiných jazycích, např. PL/I, FORTRAN a ASSEMBLER.

V nynější době se zabýváme nejen uvedenými možnostmi, ale i převodem na počítači řady TESLA, JSEP, ICL, SIEMENS.

```

BEGIN IF B THEN
L: BEGIN IF B THEN
      BEGIN FOR L:=P DO P
      END ELSE
      IF B THEN
      BEGIN IF B THEN P;
            L:=P
      END ELSE
      IF B THEN L:=P
      END ELSE
      BEGIN IF B THEN P ELSE
            IF B THEN L:=P;
            IF B THEN L:=P;
            P
      END;
L:
END
R
    
```



[1] ČSN 36 9030 Značky vývojových diagramů pro zpracování informací.
 [2] MÜLLER, H. J. — SMRČINA, J. — VOKURKA, J.: Vývojové diagramy. „Výběr z organizační a výpočetní techniky“, č. 4 a 5 1973, č. 1/1974.
 [3] NADŘCHAL, J.: Generování vývojových diagramů pomocí po-

čítače TESLA 200. Sborník Počítačová grafika FEL ČVUT, Praha 1974.

[4] Výzkumná zpráva 209-1-76 VŠSE, Plzeň.

[5] SKÁLA, V. — HLAVNÍČKA, H.: Kreslení vývojových diagramů počítačem I. „Automatizace“, 21, č. 3, s. 70.