

R-technologie pro PC

1. Úvod

R-technologie je programovací technologie pro interaktivní návrh a ladění programů. Je založena na použití grafických prostředků pro zápis řídicích programových struktur. K tomuto účelu používá tzv. R-grafy.

Pro grafický styl programování je charakteristické to, že zápis programu využívá dvourozměrnou plochu. Struktura programu je díky tomu daleko lépe členěna. Při grafickém zápise navíc není třeba verbální popis programových struktur, takže text programu je kratší a přehlednější (je-li vhodné navržen).

Použití R-technologie zvyšuje přehlednost řídicí struktury programu, vede k úspoře místa na záznamových médiích a výrazně zkracuje dobu návrhu a ladění.

V minulém roce jsem zde na semináři referoval o systému, pomocí něž je programová podpora R-technologie realizována v operačním systému DOS4. (Tento systém ve spolupráci s autory R-technologie vyvinul VOMS.) Tentokrát se soustředím na programovou podporu R-technologie na osobních počítačích.

Nejprve zopakuji základní filosofii R-technologie a uvedu stručný popis jejích prostředků.

2. Zápis programu pomocí R-grafů

Program je možné napsat jako libovolnou kombinaci textu v podobě programových příkazů a R-grafů.

R-grafy jsou grafy, jejichž vrcholy odpovídají stavům programu (resp. stavům jeho výpočtu). Významnou informaci nesou hrany grafu. Každé hrané může být přiřazena podmínka a blok příkazů. Přejod z jednoho vrcholu do druhého bude realizován, existuje-li mezi nimi hrana, pro kterou platí, že je splněna podmínka přiřazená této hraně. Při přechodu je vyplněn blok příkazů, přiřazený této hraně.

Tim, že je možné jedné hraně přiřazovat blok příkazů, dochází k prostorové efektivnímu a přitom přehlednému zápisu programu. R-grafy jsou dobře uzpůsobeny pro zobrazení na terminálu i na tiskárně díky "pravoúhlému" vedení hran grafu. Přitom u personálních počítačů je obraz programu díky lepším terminálům značně kvalitnější než u klasických alfanumerických displejů.

Používané typy grafických struktur jsou uvedeny na obrázku 1. Základním prvkem všech grafových struktur je vodorovná hrana, které je přiřazena podmínka, zapisující se nad hranu a blok příkazů, zapisující se pod ní (obr.a).

Z vrcholu A do vrcholu B přejdeme, jestliže je podmínka (P) splněna. Při přechodu se realizuje blok příkazů (S). Tato hrana tedy odpovídá případu jednoduché selekce. Nepodmíněný příkaz z ní vznikne vynecháním podmínky.

Vodorovné hrany grafu můžeme sdružovat pomocí svislých hran, ty však mají pouze pomocný význam. Na obrázku b je znázorněna několikanásobná selekce. Vodorovné hrany popisují možné přechody z A do B. Vyhodnocují se postupně shora dolů. Je-li podmínka přiřazená některé z nich splněna, provede se blok příkazů, přiřazený příslušné hraně a výpočet přejde do vrcholu B.

Směr šipek některých vodorovných hran v příkazu několikanásobné selekce můžeme obrátit - tím vznikne cyklus podobný typu repeat (obr. c). Platí, že nejprve musí být uvedeny všechny hrany vedoucí z A do B a po nich obrácené hrany. Ve vrcholu B se nejprve testuje, zda není splněna některá z podmínek na hranách vedoucích do A. Jestliže nikoliv, pokračujeme do bodu C.

Na obrázku d je uvedena druhá základní grafová struktura používaná v R-technologie. Pomocí dvojité hrany realizujeme přechod do toho samého vrcholu (např. z A do A). Tento způsob zápisu se používá, aby bylo možné udržet přehledný a "pravoúhlý" zápis grafu. Výkonné akce se zapisují na vodorovné hrany uvedené pod dvojitou hranou. Vždy musí být aspoň jedna vodorovná hrana přítomná.

Uvedená struktura může být navíc označena zvláštním návěštím jako speciální. To znamená, že se bude zpracovávat určitým, v příslušném jazyce definovaným způsobem (Např. písmenem f označíme strukturu realizující v Pascalu cyklus typu for.)

V neodladěném programu nebo programu, který zpracovává nekorektní data se při průchodu grafem můžeme dostat do vrcholu, z něhož nebudeme mít žádné korektní pokračování. Ve verzi pro CMS a DOS4 se tato situace řeší pomocí tzv. systémového východu (na předem definované návěšti), který nás upozorní na vzniklou chybu. To umožňuje velmi silnou kontrolu správnosti výpočtu. Ve verzi pro PC to bohužel tak není. V případě kolize se počítá s nuceným přechodem do dalšího vrcholu.

3. Programová podpora R-technologie na PC

System pro podporu R-technologie na PC byl zatím realizován jako nadstavba Turbo systémů pro jazyky C a Pascal a jako vlastní systém (s analogickou "turbo"-filosofií) pro Modulu 2. Je to systém, který programátoru umožňuje psát programy grafickým stylem a pohybovat se přitom v pohodlném kompaktním integrovaném prostředí.

Skládá se ze tří částí, které se spouštějí odděleně.

Jsou to:

- grafický editor,
- grafický debugger,
- generátor a analyzátor cest.

Editor

Při vyvolání editoru například grafického Pascalu se objeví rozšířené menu Turbopascalu, umožňující editovat a překládat programy napsané pomocí grafického jazyka.

Na horním kraji obrazovky se objeví základní menu grafického editoru. Můžeme volit z pěti základních možností (režimy Input, Edit, Special, Service, Program).

Podívejme se blíže na jednotlivé režimy.

Input umožňuje vstup nových grafických struktur. Je možné zvolit jednu ze dvou základních grafických struktur (zvedených v části 2), přidávání dalších vodorovných hran k nim (a jejich orientaci a umístění vzhledem k pozici kurzoru, a vkládání grafické struktury uložené v bufferu (pokud je ovšem naplněn). Dostupné jsou přitom pouze základní možnosti, ostatní jsou zakódovány. V menu je od sebe snadno rozpoznáme, protože jsou vyznačeny různými barvami (v základním barevném provedení je k jejich zobrazení použita černá a bílá).

Režim Edit je určen k odstraňování nepotřebných částí

Režim Special slouží k třem účelům. Za prvé k označení speciálních programových struktur definovaných v příslušném grafickém programovacím jazyce (case, cyklus, typu for ap.), za druhé k popsání tzv. strukturovaných skoků (typu předčasně ukončení provádění programové struktury ap.) a za třetí k obsluze bufferu, v němž se ukládají grafické struktury, které pak můžeme v režimu input přenášet na jiné místo v programu.

Režim Service obsahuje skupinu několika mírně neaouro-
dých akcí. Patří do nich ukončení práce s editorem (s uložením resp. bez uložení editovaného souboru), hledání řetězce podle zadaného vzoru, zobrazení základních informací o grafu (kolik mají hran ap.) a nastavování barev pro zobrazení grafů.

Režim Fragment umožňuje pracovat s označenými fragmenty programu (označení fragmentu, jeho přesun, kopírování, tisk fragmentu, kopírování fragmentu do jiného souboru).

Po návratu k menu rozšířeného Turbopascalu lze vyvolat torbopřeklad programu napsaného grafickým způsobem.

Debugger

Debugger nabízí režim prohlížení grafického zápisu programu (a nastavením ladících bodů) a tři ladící režimy.

Při ladění je možné postupovat po krocích (z vrcholu do vrcholu), do nejbližšího nastaveného ladícího bodu nebo zvolit automatický režim, v němž výpočet probíhá nad grafem bez přerušení s nastavováním rychlosti výpočtu pomocí čísel 1, 2 až 0 (zastavuje se pomocí Enter).

Při ladění je možno definovat okénko dat a nastavit je na data, jejichž hodnotu chceme během výpočtu sledovat (v tomto okénku je též možné nastavovat podmínky zastavení

související s uvedenými daty), prohlížet obsah registrů a vyvolat okno s fyzickými daty na úrovni paměti (které lze upravovat).

Dále je možné nastavit režim zápisu tras, které jsme při ladění procházeli, pro další analýzu a rozhodnout se, zda při ladění máme vcházet do volaných procedur (tj. přecházet na jejich grafy) nebo ne.

Laděný program můžeme v případě potřeby restartovat se zachováním nastavených ladících bodů a podmínek.

Modul pro testování a analýzu

Tento modul v režimu testování generuje ladící cesty v programu, který je napsaný pomocí R-grafů. Umožňuje jednotlivé cesty zobrazit na grafu a současně generuje buď seznam všech vrcholů, kterými cesta prochází nebo seznam podmínek, potřebných pro průchod příslušnou cestou. Umožňuje rovněž zobrazit seznam všech vygenerovaných cest.

Pro generaci cest jsou použita dvě různá kritéria - maximálního pokrytí a lineárně nezávislých cest (podle McCabe).

V režimu analýzy se pak, jestliže jsme při ladění zvolili režim zápisu trasovaných cest, zobrazí cesty a hrany, které zůstaly při ladění nepokryté.

Zatímco pro Turbo C a Turbo Pascal již existují kvalitní debuggery srovnatelné s debuggerem popsaným výše, prostředek podobný tomu, který jsme právě uvedli, nebývá obvykle k dispozici. Snadnost a průzračnost jeho použití spočívá v grafickém zápisu programu, nad nímž lze testovací cesty velmi snadno a přehledně vizualizovat.

4. Hodnocení uvedeného systému

To, co jsme právě popsali, je nástin ideálního systému, který je momentálně k dispozici pro Modulu 2, zatímco pro Pascal a C je komerčně přístupný zatím pouze grafický editor.

Zbývající dva moduly jsou však v podstatě hotové, přecházejí zkušební fázi. Do konce roku by tedy měly být plně k dispozici i u nás, v ČSSR.

Je zajímavé porovnat, jakým způsobem se vyvíjela filozofie návrhu programových prostředků pro podporu technologie, od její první verze, vzniklé pro OS, přes verze pro CMS a DOS až k verzi pro PC.

Na začátku šlo o ideu realizace zvláštního virtuálního R-procesoru, který by byl prostředkem pro budování dalších, uživatelských nadstaveb, užitých na airu konkrétních projektů.

Ve verzi pro CMS se od této představy upustilo. Programový systém na podporu R-technologie byl budován jako silná podpora strukturovaného návrhu programu shora dolů. Při návrhu pak byl velmi efektivně využit grafický zápis programu. Dá se říci, že v této verzi jsou prostředky pro podporu návrhu programu a pro usnadnění a zvýšení přehlednosti samotného programování v rovnováze.

Verze pro PC se jednoznačně orientuje na zkvalitnění grafického stylu programování. Je to spíše grafický preprocesor (velice efektivní) než programovací technologie.

Uvedený vývoj má pochopitelně kladnou i zápornou stránku. Je škoda, že se prakticky vytrácí podpora návrhu programu. Na druhé straně z toho však vyplývá jednoduchost a efektivnost verze R-technologie na PC.

Nelze pochybovat, že jde o přínos opravdu markantní. Grafický styl programování, který R-technologie na profesio-

nální úrovni podporuje, má značné výhody. Podívejme se blíže na hlavní z nich.

V programování jsou stále otevřeny otázky čitelnosti programů, způsobů jejich testování a zejména jejich dokumentace.

Programátoři většinou svou práci začínají tím, že si nakreslí nějaké schéma programu. Program a jeho algoritmy však při ladění doznají mnoha změn, takže původní schéma je pro potřeby dokumentace prakticky nepoužitelné. Bylo by vhodné je aktualizovat, jenže to je jev, ke kterému, ať už se na tom podílí podcenění této úlohy, nechť, časový tlak nebo lenost, dochází sřídkakdy.

Proto se různé metodiky, technologie a pravidla programování snaží programátory všelijak ukáznit a přidržet u určitých zásad, které usnadňují údržbu programů a prodlužují jejich životnost. Většinou nepřilíš úspěšně. Brání v tom složitost osvojení si potřebných prostředků a hlavně pracnost jejich použití.

To při použití R-technologie do značné míry odpadá. Vzhledem k jednoduchosti R-grafů se základům grafického programování lze naučit nasmírně rychle. Samotný program napsaný pomocí R-grafů má velmi průzračnou logiku. K jeho pochopení stačí znát obecný popis algoritmu a rozumný popis dat.

R-grafy vlastně řeší otázku autorské dokumentace a je údržby do značné míry samy. Jsou stejně čitelné jako vývojové diagramy nebo strukturogramy. Zdrojový text zapsaný pomocí R-grafů je součástí autorské dokumentace. Přitom jakékoliv úpravy programu se do grafického zápisu automaticky promítají. Po odladění tak máme k dispozici i "bezbolestně" získanou základní část dokumentace a to v aktuálním a správném tvaru.

Uveďme jako příznačnou výpověď jedné programátorky.

kteřá začala používat R-technologie:

"Ukázali mi výtisk cizího programu, který jsem viděla poprvé. Během pěti minut mi řekli základní principy R-technologie, dvacet minut mi vysvětlovali, co program dělá a ještě mi během deseti minut zadali úkol, co nám do programu dodělat.

Program byl napsán v grafickém Pascalu. Do té doby jsem programovala pouze v jazycích PL/I, Fortran a Assembler. Dva dny jsem strávila poznáváním cizích algoritmů (přitom program skoro nebyl komentován). Čtyřikrát jsem byla nucena obrátit se na autora programu s dotazy na použití proměnných. Kdyby byly proměnné komentovány, nebyly by dotazy nutné. Třetí den jsem začala měnit cizí algoritmy. Nic podobného jsem při programování v lineárních programovacích jazycích nezažila."

Další výhodou grafického programování (kromě vyšší přehlednosti a názornosti zápisu programu) je i to, že programové řídicí struktury jsou realizovány pomocí grafů. V jejich zápise se v klasicky psaných programech často objevují chyby, které se obtížně hledají. To při programování pomocí grafů odpadá.

Nezanedbatelnou výhodou je i to, že grafický tvar řídicí struktury vyžaduje nepoměrně menší práci při jejím zápisu. Podívejme se na následující tabulku:

jazyková	minimální počet úderů	
konstrukce	text	R-graf
FOR..TO..DO..END	10	4
WHILE..DO..END	10	3
IF..THEN..ELSEIF	22	5
THEN..ELSE..END		

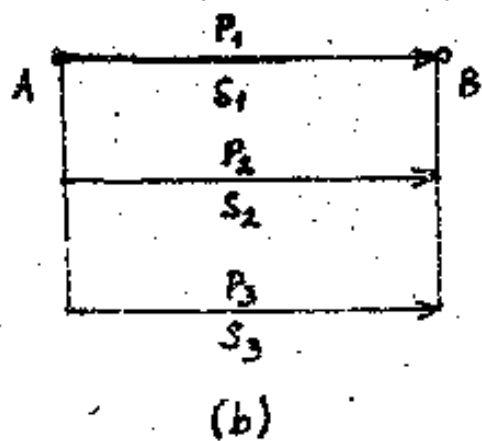
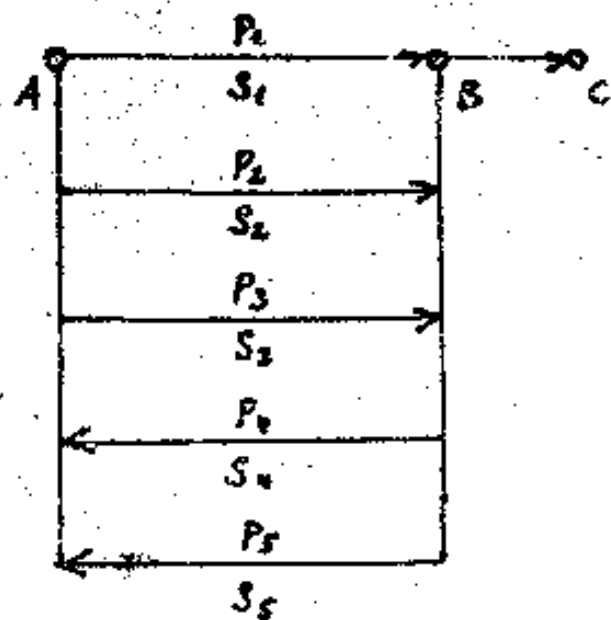
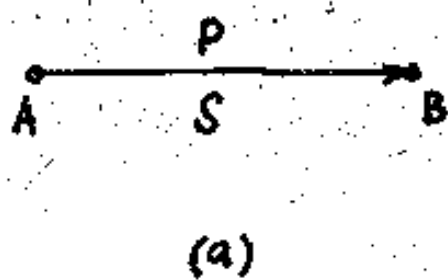
Navíc právě programové struktury se od sebe v jednotlivých programovacích jazycích nejvíce odlišují. Učení se způsobům, jak je správně používat, zabírá značnou část činnosti programátorů, učících se nový jazyk.

Zápis podmínek, příkazy přiřazení a volání procedur, které se v R-technologii zapisují na hrany grafu, jsou ve většině algoritmičtějších programovacích jazyků podobné. Proto, jestliže osládnete programování v grafickém Pascalu, nebude Vám dleto velké potíže přejít na programování v grafické PL/I nebo Modula.

6. Závěr

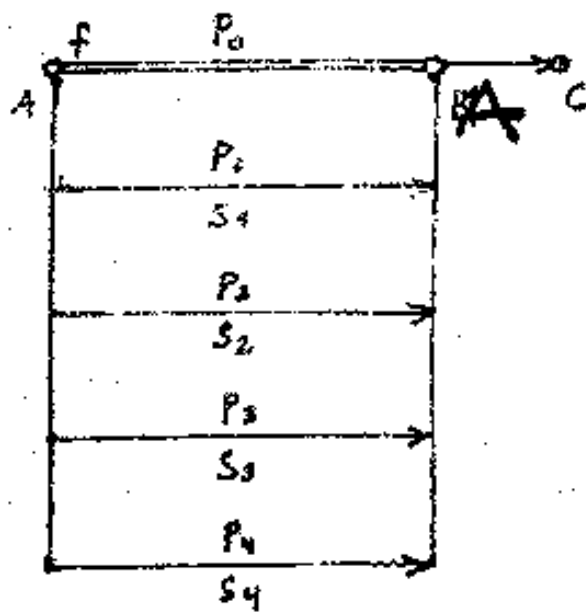
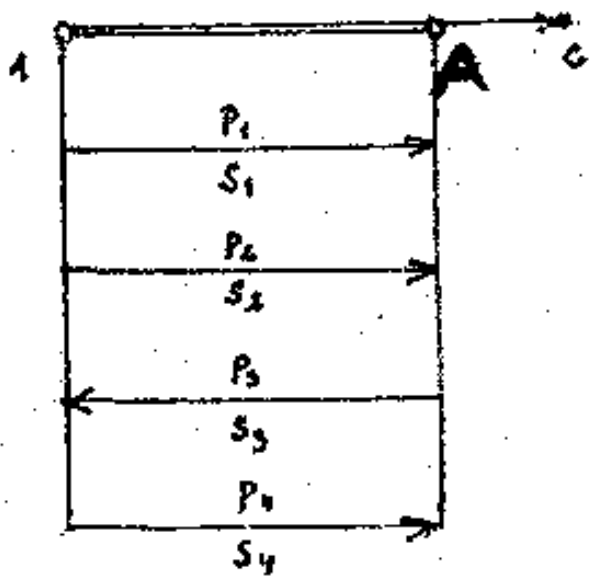
V současné době se mnoho hovoří o vizualizaci programování. Je mnoho různých technologií a forem grafické reprezentace programů a R-grafy jsou jen jednou z nich.

Jejich jednoduchost, názornost, nezávislost na jazycích, ekonomická výhodnost a snadnost realizace jejich programové podpory je zřetelně vyvýšují nad úroveň ostatních známých prostředků.



(c)

(b)



(d)

(e)

obrázek 1