

Algoritmus a jeho vlastnosti, špecifikácia algoritmickej úlohy

Algoritmus je postup riešenia nejakej úlohy. Pozostáva z príkazov na vykonávanie takých činností, ktoré jeho vykonávateľ dokáže urobiť. **Program** je algoritmus zapísaný v nejakom programovacom jazyku. Programy v počítači sú určené pre procesor, ten je ich vykonávateľom.

Ak máme zostaviť nejaký program, potom vlastne musíme zostaviť algoritmus, teda vyriešiť zadanú algoritmickú úlohu. Riešením algoritmickej úlohy získavame nové informácie. Keďže sú produktom riešenia, hovoríme im **výstupné informácie** alebo **výstupné údaje**. Informácie, z ktorých pri riešení úlohy vychádzame, nazývame **vstupné informácie** alebo **vstupné údaje**.

Pomocou vstupných a výstupných údajov zadávame úlohu na riešenie, určujeme, akú úlohu ideme vlastne riešiť. Vstupné údaje poznáme pred riešením v explicitnom tvare – ako konkrétne údaje. Výstupné údaje pred riešením v konkrétnom tvare nepoznáme. Výstupné hodnoty určujeme implicitne – skryto, pomocou podmienky, ktorú musia spĺňať. Ak máme napríklad úlohu, v ktorej treba nájsť najväčšieho spoločného deliteľa dvoch celých kladných čísel A, B, vstupnými údajmi budú dané dve celé kladné čísla A, B ako konkrétne hodnoty. Výstupnú hodnotu určujeme pomocou podmienky, že to bude najväčšie celé číslo, ktoré delí A aj B bez zvyšku.

Podmienku, ktorú musia spĺňať výstupné údaje nazývame **výstupná podmienka**.

Ani vstupnými údajmi nemôžu byť ľubovoľné hodnoty, ale musia vyhovovať istej podmienke, ktorú nazývame **vstupná podmienka**.

Vstupnou a výstupnou podmienkou charakterizujeme úlohu, ktorú máme riešiť, špecifikujeme to, čo treba riešiť. Nejde nám o jednu konkrétnu úlohu, ale o celú triedu úloh, ktorých vstupné údaje spĺňajú vstupnú a výstupné údaje výstupnú podmienku. Pretože na riešenie takejto úlohy chceme vytvoriť algoritmus, nazývame ju algoritmická úloha.

Skutočnosť, že úlohu máme špecifikovanú, že vieme čo máme riešiť, ešte neznamená, že sa nám musí vždy podariť vytvoriť zodpovedajúci algoritmus na jej riešenie. Existujú úlohy, pre ktoré to nie je možné. Akékoľvek úvahy o riešení úlohy však možno robiť iba vtedy, ak vieme, čo treba riešiť.

Keďže špecifikácia úlohy platí pre ľubovoľnú úlohu danej triedy, vstupnými a výstupnými údajmi sú premenné, ktoré pre konkrétne úlohy nadobudnú konkrétne hodnoty. Vstupná a výstupná podmienka sa teda týka práve hodnôt premenných.

Príklad 1.:

Úlohu výpočtu obsahu štvorca, povrchu a objemu kocky so stranou a možno špecifikovať takto:

Vstupná premenná: **a**

Vstupná podmienka: **a** je celé kladné číslo

Výstupné premenné: **o** (obsah štvorca), **p** (povrch kocky), **v** (objem kocky)

Výstupná podmienka: **o** = a^2 , **p** = $6 \cdot a^2$, **v** = a^3

Príklad 2.:

Úlohu hľadania najväčšieho spoločného deliteľa čísel a , b možno špecifikovať takto:

Vstupné premenné: a , b

Vstupná podmienka: a , b sú celé kladné čísla

Výstupné premenné: NSD

Výstupná podmienka: NSD delí a aj b bez zvyšku, pričom neexistuje spoločný deliteľ väčší ako NSD, ktorý delí bez zvyšku a aj b .

Príklad 3.:

Úlohu výpočtu koreňov kvadratickej rovnice možno špecifikovať takto:

Vstupné premenné: a , b , c

Vstupná podmienka: a , b , c sú celé kladné čísla

Výstupné premenné: x_1 , x_2

Výstupná podmienka: $a \cdot x_1^2 + b \cdot x_1 + c = 0$, $a \cdot x_2^2 + b \cdot x_2 + c = 0$

Po špecifikácii úlohy, po špecifikácii toho, čo treba riešiť, pristupujeme k tomu, ako ju riešiť, teda k tvorbe postupu riešenia. Tu už treba brať do úvahy nielen samotnú úlohu, ale aj realizátora – procesor postupu riešenia. Každý procesor je schopný vykonávať určitý počet akcií – operácií, pričom každá akcia má predpísané podmienky realizácie, účinok realizácie a konečný čas trvania. Čo znamená utvoriť postup riešenia? Najprv sa pozrieme, či sa dá úloha riešiť použitím niektorej z akcií procesora. Ak áno, použijeme ju a úlohu máme vyriešenú. Ak nie, musíme úlohu rozkladať na podúlohy, a to dovedy, kým nezodpovedajú podúlohám, ktoré vieme niektorou z akcií procesora riešiť. Kombináciou – kompozíciou akcií zodpovedajúcich jednotlivým podúlohám dostaneme hľadaný postup riešenia. Jeho použitím na vstupné údaje zabezpečíme ich transformáciu na výstupné údaje – údaje spĺňajúce výstupnú podmienku.

Ak zoberieme úlohu z príkladu 1. a procesor vie násobiť celé čísla, tak postup riešenia je priamočiary. Stačí použiť výstupnú podmienku, v ktorej umocnenie nahradíme násobením. Úlohy v príkladoch 2. a 3. sú už zložitejšie. Ich výstupné podmienky už postup riešenia neobsahujú.

Akcie, ktoré je procesor schopný vykonávať sú zvyčajne jednoduché a transformáciu vstupných údajov na výstupné treba zabezpečiť ich kompozíciou. Transformácia sa teda neuskutočňuje v jednom kroku, ale postupne, ako sa vykonávajú jednotlivé akcie procesora. Každá akcia mení hodnoty – stav premenných vyskytujúcich sa v algoritme: Postup riešenia predpisuje následnosť vykonávania jednotlivých akcií. Všeobecné pravidlá určujúce postupnú transformáciu vstupných údajov na výstupné nazývame algoritmus.

Pre algoritmus, ako postup riešenia úloh platia tzv. základné vlastnosti: hromadnosť, determinickosť a rezultatívnosť.

Pod **hromadnosťou** rozumieme skutočnosť, že algoritmus nie je použiteľný iba na jednu konkrétnu úlohu, s konkrétnymi vstupnými údajmi, ale na ľubovoľné vstupné údaje spĺňajúce vstupnú podmienku.

Determinickosť algoritmu znamená, že v každom kroku algoritmu musí byť jednoznačne dané, čo robiť. Realizácia algoritmu nesmie byť podmienená ďalšími podmienkami ako tými, ktoré sú v ňom uvedené.

Rezultatívnosť algoritmu znamená, že proces predpísaný algoritmom je konečný, skončí sa po konečnom počte krokov. Niekedy hovoríme stručne, že algoritmus je konečný.

Overenie hromadnosti a determinickosti je pomerne ľahké. Oveľa zložitejšie sa overuje rezultatívnosť algoritmu. Rezultatívnosť, resp. konečnosť algoritmu je spojená s ďalšou vlastnosťou algoritmu, a to **správnosť** algoritmu. Správnosť algoritmu určujeme vzhľadom na vstupnú a výstupnú podmienku.

Algoritmus je vzhľadom na vstupnú a výstupnú podmienku správny vtedy, ak pre všetky vstupné údaje spĺňajúce vstupnú podmienku sa proces predpísaný algoritmom zastaví (konečnosť algoritmu) a výstupné údaje spĺňajú výstupnú podmienku.

Overenie správnosti algoritmu patrí medzi najdôležitejšie činnosti spojené s tvorbou algoritmov. Ak algoritmus nie je správny, ak nerieši našu úlohu, tak bola celá naša činnosť zbytočná. Z definície správnosti algoritmu vyplýva aj dôležitosť správnosti špecifikácie úlohy. Darmo utvoríme správny algoritmus, ak špecifikácia, na základe ktorej sme dokázali jeho správnosť, nezodpovedá našej úlohe. To, že algoritmus dáva správne výsledky pre jednu konkrétnu vstupnú hodnotu ešte neznamená, že je správny. Na to potrebujeme ukázať, že dáva správne výsledky pre všetky vstupné údaje spĺňajúce vstupnú podmienku.

Algoritmus možno definovať tiež ako konštruktívne zobrazenie množiny n -tíc vstupných údajov do množiny m -tíc výstupných údajov. Konštruktívnosťou vyjadrujeme skutočnosť, že algoritmus udáva ako určiť k daným vstupným údajom zodpovedajúcu výstupné údaje. Z hľadiska samotného zobrazenia môže k jednému zobrazeniu existovať viac algoritmov, viac spôsobov ako určiť zodpovedajúce výstupné údaje k daným vstupným údajom. Algoritmy, ktoré určujú rovnaké zobrazenie sa nazývajú **ekvivalentné**.