

Alternatyvūs algoritmavimo mokymo būdai

Valentina DAGIENĖ (MII)

el. paštas: dagiene@ktl.mii.lt

1. Įvadas

Nuo 1999–2000 mokslo metų informatikos kursas perkeltas į pagrindinę mokyklą, jį privaloma dėstyti IX–X klasėse. Rerūiantis bendrąja informatikos programa [7] ši kursą sudaro keturios dalys, iš kurių viena skirta algoritmavimui mokyti. Iš viso informatikos privalomam kursui IX–X klasėse skirta 68 val. Formaliai paskaičiavus algoritmams tektų 1/4 dalis, t. y. tik 17 valandų.

Išmokyti perprasti algoritmus – nėra lengva. O jei dar sieksime, kad mokiniai ne tik gebėtų perskaityti pateiktą algoritmą, bet ir patys sudarytų kad ir nedidelio uždavinio algoritmą – daugeliui besimokančių gana sunki problema.

Ką daryti? Kaip mokyti algoritmavimo pagrindinėje mokykloje?

Norėdami rasti atsakymus į šiuos klausimus analizuosime dažniausiai pasaulyje vartojamus būdus algoritmams mokyti. Pasiremdami analizės rezultatais pateiksime ir įvertinsime galimus sprendimus.

2. Algoritmai mokykloje: mokyti ar nemokyti?

Klasika tapęs klausimas keliamas nuo tada, kai tik informatika atsirado bendrojo lavinimo mokyklose. Lietuvoje nuo 1986 m. iki pat pertvarkos 1999 m. privalomas informatikos kursas buvo dėstomas X–XII klasėse (nepagrindinėje mokykloje), didžiausia dalis buvo skirta algoritmavimui, dėl kompiuterių stygiaus dažnai buvo mokoma tik teoriniu lygiu. Taip darė ir kitos šalys, ypač kurių mokyklų programos orientuotos į fundamentalių žinių suteikimą.

Algoritmavime siekiama išmokyti suprasti formalių žymenų reikšmę, gebėti juos vartoti atitinkamiems veiksams aprašyti. Šio kurso dalis glaudžiai siejasi su matematika.

Be abejo, algoritmams suprasti reikia nemažai matematinių žinių. Tačiau matematikoje labiau akcentuojamas veiksnių formalizavimas, sprendimo būdų įvaldymas, paprastai dėmesys kreipiamas į atskiras matematinių uždavinių sprendimo komponentes.

Algoritmavime svarbią vietą užima ne tik veiksnių, bet ir duomenų, aprašų, pasirinktos kalbos formalizavimas, produkto užbaigimas ir pateikimas. Kreipiamas dėmesys į formalios kalbos konstrukcijų sampratą, uždavinio algoritmo atlikimą kompiuteriu.

UNESCO pateiktose informatikos mokymo rekomendacijose [6] algoritmų siūloma mokyti pasirinktiniu, sustiprintu lygmeniu. Daugelyje kitų dokumentų, pavyzdžiui, IFIP

darbo grupės WG 3.1 parengtose rekomendacijose [9, 10] kaip tik siūloma visoms šalims rūpintis algoritminio mąstymo ugdymu.

Algoritmavimo mokymas priklauso nuo šalies politikos (kokių žinių ir gebėjimų siekiama suteikti bendruoju lavinimu), tradicijų (integruoti ar atskiri dalykai, kokiais metodais mokoma), mokytojų kvalifikacijos ir t. t. Lietuva turi ilgametę algoritmavimo mokymo patirtį, gajas sąsajas su taikomosios matematikos uždaviniais, nemažą pedagogų bei mokslininkų įdirbį šioje srityje. Taip kad neturėtų kilti abejonių, jog dalis informatikos kurso skiriama algoritmams mokyti. Tik svarbiausia – kaip mokyti? Kaip pateikti kursą, kad daugelis moksleivių įvaldytų būtinausias sampratas ir perprastų darbo kompiuteriu principus?

3. Svarbiausios algoritminės konstrukcijos

Pagrindinės procedūrinio programavimo sąvokos yra duomenų tipai ir valdymo struktūros. 1986–1998 m. informatikos kurse buvo supažindinama su duomenų tipais: sveikuoju, realiuoju, loginiu, masyvu (sustiprintam lygiui) bei valdymo struktūromis: priskyrimo, sąlyginio sakiniams, žinomo ir nežinomo kartojimų skaičiaus ciklais, procedūra, funkcija.

Perkėlus informatikos kursą į pagrindinę mokyklą, algoritmavimą stengtasi supaprastinti. IX–X klasėse vartojami tik trys duomenų tipai (sveikasis, realusis, loginis) ir pagrindinės valdymo konstrukcijos (priskyrimo ir sąlyginis sakinytis bei dviejų rūšių ciklai). Kadangi atsakyta procedūrų ir funkcijų (algoritmas apiforminamas programa), tuomet nagrinėjami rašymo ir skaitymo sakiniai.

XI–XII klasėse algoritmavimo mokoma tik realiniame profilyje, bendrajam kursui skirtas vienas iš dviejų modulių – programavimo pradmenų [7]. Jame iš dalies pakartojamas IX–X klasės kursas (per praktinių uždavinių sprendimą) ir nagrinėjamos naujos algoritminės konstrukcijos – duomenų tipai (masyvas, įrašas) bei valdymo struktūros (procedūra, funkcija). Supažindinama su tekstinėmis bylomis – dideliems pradinėms duomenų rinkiniams skaityti bei rezultatams rašyti.

4. Paskalio kalba

Algoritmų mokyti galima įvairiais metodais: vartojant formalias kalbas, vaizdines schemas, kompiuterizuotas programas ir pan.

1970 m. N. Wirth'ui sukūrus Paskalio kalbą ji pasirodė viena tinkamiausių algoritmams užrašyti. Smarkiai paplito ir dabar tebe naudojama daugelyje šalių.

Lietuvos mokyklose nuo pat informatikos atsiradimo algoritmus buvo siūloma mokyti Paskaliu, tuo rėmėsi daugelį metų mokyklose vartojamas vadovėlis [3]. Reikia atkreipti dėmesį, kad algoritmavimui mokykloje mokyti vartojama tik dalis Paskalio kalbos, kita (gilesnė) sistemos dalis yra skirta profesionaliam programavimui.

Reformuotos pagrindinės mokyklos informatikos bendrojoje programoje algoritmams mokyti nesiūloma jokių konkrečių priemonių. Tačiau parengta vadovėlio dalis [4] pagrįsta Paskalio kalbos minimalių konstrukcijų vartojimu. Čia dėstomi sveikieji ir realieji skaičiai, loginis tipas, priskyrimo bei sąlyginis sakiniai, ciklai.

5. Algoritmų vaizdavimas grafiškai

Algoritmams mokyti neretai pasitelkiama grafinių priemonių – blokinių schemų, struktūrogramų ir pan. Tai palengvina suvokti algoritmavimo konstrukcijomis užrašytus veiksmus. Tačiau paprastai vaizdinės schemos vartojamos greta kurios nors programavimo kalbos, o nėra vien algoritmų pateikimo forma.

Struktūrogramos vartojamos ir IX–X kl. vadovėlyje [4]. Ypač jomis palanku aprašyti realių taikomųjų uždavinių sprendimus, kuriems reikėtų sudėtingesnių Paskalio konstrukcijų.

Šiuo metu pasaulyje pastebima tendencija kurti kompiuterizuotas algoritmų vaizdavimo schemas. Ypač patraukliai atrodo algoritmų surinkimas iš komponentų – tarytum konstravimas iš Lego kaladėlių [8].

Šitoks algoritmavimo būdas turėtų būti kur kas patrauklesnis mokykloms. Juo šiuo metu smarkiai domimasi pasaulyje, Lietuvoje šia linkme beveik dar nedirbama.

6. Logo – alternatyva Paskaliui

Daugelis informatikos mokytojų teigia, kad IX–X klasėse per sunku dėstyti algoritmavimą vartojant Paskalio programavimo kalbą. Didesnę dalį priežasčių sąlygoja pati programinė įranga – ji nėra draugiška pradedančiajam, be to mūsų šalyje naudojamas Turbo Paskalis remiasi senstelėjusia DOS operacine sistema, varduose negalima vartoti specifinių lietuviškų raidžių ir pan. Modernesnės Paskalio kalbą vartojančios sistemos yra per brangios mokykloms.

Logo kalbą realizuojanti „Komensio Logo“ sistema nupirka visoms Lietuvos mokykloms, ja dirbama būreliuose, kai kurių mokyklų jaunesnėse klasėse. Tad ar nebūtų galima panaudoti Logo pagrindiniams algoritmavimo įgūdžiams lavinti? Pasaulyje esama tokios patirties [2]. Šitai daro Slovakijos, Lenkijos, Rusijos mokyklos.

„Komensio Logo“ turtingesnė ir tiktamojo programavimo galimybėmis, ji visiškai suderinta su Windows operacine sistema, draugiška vartotojui [1]. Pradedantiesiems svarbu kuo greičiau pamatyti algoritmų duodamą rezultatą – tuo neabejotinai Logo kalba pranašesnė: komandas vykdo dinaminis objektas (vėžliukas), jos interpretuojamos ir tuoj pat pateikiamas rezultatas (ne tik skaičiai, bet ir grafika, tekstai, langų, mygtukų formos ir pan.).

7. Pagrindinių algoritmavimų užrašai Logo kalba

Peržvelkime, kaip būtų užrašomos pagrindinės algoritmavimo konstrukcijos, vartojamos IX–X klasių informatikos kurse, Logo kalba.

Duomenų tipai. Logo kalboje nėra privaloma griežtai aprašyti duomenų tipų, kas moksleiviams paprastai būna sunkiai suvokiama. Sveikieji ir realieji skaičiai vartojami natūraliai. Loginio tipo išreikštiniu pavidalu nėra. (Apie duomenų tipus Logo kalboje išsamiau straipsnyje [5]).

Kintamieji ir reikšmių priskyrimas. Kintamojo sąvoka pradedančiajam programuoti sunkiai suvokiama. Paskalio kalboje kintamojo atributus (vardą, tipą, reikšmę) nusako vardo aprašas, tipo apibrėžtis ir priskyrimo sakiniu (arba skaitymo) suteikiama reikšmė. Tipo apibrėžčių pradedantiems galima nepateikti, tačiau kintamųjų aprašai ir reikšmės būtinos.

Logo kalboje ta pačia komanda suteikiamas kintamajam vardas ir reikšmė, pavyzdžiui, tobus "a 15. Tai aišku ir paprasta.

Logo kalboje paprastai ir vaizdžiai sprendžiama vidinių ir išorinių (lokalijų ir globalijų) kintamųjų problema: norėdami aprašyti vidinį kintamąjį (procedūroje) vartojame komandą tobus "a 15 (kintamasis a paskelbiamas vidiniu ir tuo pačiu jam priskiriama reikšmė). Yra ir kita galimybė kintamajam tapti vidiniu: vartojant specialią Logo komandą vidinis.

Sąlyginis sakiny. Logo kalba turi net kelių rūšių sąlyginius sakinius (jeigu, jei, tiesa, jei, melas). Bendriausias sąlyginis sakiny yra

jei <sąlyga> [komandų sąrašas] arba

jei <sąlyga> [komandų sąrašas 1] [komandų sąrašas 2]

Tai visiškai analogiški Paskalio kalbai žymenys (nėra tik priešingus veiksmus nurodančio žodžio else).

Ciklai. Ciklo esmę perteikti nėra paprasta, todėl gerai būtų, kad formalus jo užrašas būtų kuo paprastesnis ir vaizdesnis. Vargu, ar begalima daugiau supaprastinti ciklo užrašą negu kad yra Logo kalboje:

kartok n [komandų sąrašas].

Šis ciklas akivaizdus – laužtiniuose skliaustuose esančias komandas reikia kartoti n kartų.

Logo kalboje yra ir kitų ciklo konstrukcijų-komandų: kol (atitinka Paskalio kalbos while), ciklas (atitinka Paskalio kalbos for ciklą, tik dar bendresnį – galima nurodyti ciklo kintamojo žingsnio dydį), kiekvienam (darbui su sąrašais).

Procedūros. Pagrindinės mokyklos algoritmavimo kurse nėra numatyta mokyti procedūrų. Vartojant Paskalio programavimo kalbą procedūras (o ypač funkcijas) nėra lengva išaiškinti. Tačiau Logo kalba kiekvienas algoritmas aprašomas procedūra – natūraliai ir paprastai. Todėl dirbantys Logo kalba mokiniai lengvai turėtų suvokti procedūras ir gebėti jas vartoti uždaviniams spręsti.

8. Išvados

Bendrojo lavinimo mokyklos informatikos kurse nedidelė dalis skiriama pažinčiai su algoritmavimu. Algoritmų pradedama mokyti pagrindinėje mokykloje – IX–X klasėse. Labai svarbu pasirinkti tinkamą priemonę (žymenis) algoritmams užrašyti.

Iš trijų straipsnyje pateiktų būdų – Paskalio programavimo kalbos, grafinių schemų bei Logo kalbos – pradedantiems labiausiai teiktina Logo kalba.

Logo kalba lengvai ir vaizdžiai išreiškiamos pagrindinės duomenų ir valdymo konstrukcijos. Jei kintamųjų reikšmių priskyrimas ir sąlyginis sakiny dar galėtų būti beveik apyvienodžio sunkumo, tai ciklą nėra nė ko lyginti – Logo kalba tai užrašoma žymiai aiškiau. Tas pats teigtina apie procedūras bei kitas algoritmines konstrukcijas.

Literatūra

- [1] A. Blaho and I. Kalas, *SuperLogo: Learning by Developing*, Longman Logotron, Cambridge (1998).
- [2] P. Brusilovsky *et al.*, Mini-languages: A way to learn programming principles, *Education and Information Technologies*, 2(1), 3–24 (1997).
- [3] V. Dagienė, G. Grigas, *Informatika: Bandomoji mokymo priemonė X–XII kl.*, Šviesa, Kaunas (1991).
- [4] V. Dagienė, *Informatikos pradmenys: Algoritmai*, II d., TEV, Vilnius (1999).
- [5] V. Dagienė, „Komenskio Logo“ duomenų tipų analizė, *Lietuvos matematikų draugijos XXX konferencijos darbai*, Vilnius, 222–225 (1999).
- [6] *Informatics for Seconadry Education: A Curriculum for Schools*, UNESCO, Paris (1994).
- [7] *Informatika Lietuvos bendrojo lavinimo mokykloje: Bendrosios programos ir išsilavinimo standartai* // Sudarė V. Brazdeikis, Leidybos centras, Vilnius (1999).
- [8] J. Roschelle *et al.*, Developing educational software components, *Computer*, September, 50–56 (1999).
- [9] H.G. Taylor, R.M. Aiken, T.J. van Weert, *Guidelines for Good Practice: Informatics Education in Secondary Schools*, IFIP Working Group 3.1, Geneva (1993).
- [10] T.J. van Weert, *Guidelines for Good Practice: Integration of Information Technology into Secondary Education*, IFIP Working Group 3.1, Geneva (1993).

Alternative ways of teaching algorithms

V. Dagienė

This paper deals with ways of teaching algorithms in basic comprehensive school. Three main methods are described: the programming language Pascal, graphical schemes, and Logo.

The main attention is paid to the analysis of using Logo for developing algorithms in elementary informatics course for IX–X grade students. The basic algorithmical constructions: variables, assignment, conditional statements, loops, procedure are described.