

Tvorba kurikula pro výuku informatiky jako vědy pro česká gymnázia

Daniel Lessner

Katedra software a výuky informatiky, Matematicko-fyzikální fakulta, Univerzita Karlova v Praze

Malostranské nám. 25

118 00 Praha 1

Česká republika

lessner@ksvi.mff.cuni.cz

ABSTRAKT

V České republice informatika dosud není zařazena do všeobecného vzdělávání. V rámci snahy o zlepšení tohoto stavu jsme realizovali rozsáhlejší projekt, který zahrnoval vytvoření konceptu výuky úvodu do informatiky, jeho ladění, úpravy a závěrečné vyhodnocení. Ukázalo se, že jde o téma sice náročné, ale zvládnutelné a jednoznačně prospěšné.

V tomto článku popisujeme dílčí část projektu, totiž tvorbu programu výuky. Není k dispozici tradiční a osvědčené řešení, o které by bylo možné se opřít, proto jsme začali s čistým štítem a na základě identifikace fundamentálních idejí didakticky transformovali obor informatiky pro gymnázium. Vybrali jsme pojmy informace, model, algoritmus a efektivita jako ty, na nichž vzdělávání vystavět. Lze je nahlížet na různé úrovni pokročilosti, objevují se opakovaně v různých oblastech místech informatiky, přitom mají svůj význam i mimo ni. Na závěr článku krátce komentujeme související výsledky pilotáže.

ABSTRACT

Computer science is not a part of general education in the Czech Republic. We have carried out a larger project to help to improve this situation. It involved development of a new introductory computer science course concept, its iterative testing and adjusting and final piloting and evaluation. Computer science turned out to be difficult, yet manageable by our students, and computer science education showed some clear benefits.

In this article we describe a part of this project, the initial curriculum development. There was no traditional and well tested approach we could rely on. So we started with clear slate and carried out the didactic transformation of computer science for grammar school. Our approach was based on identifying fundamental ideas, such as information, model, algorithm and efficiency. These are the foundation of the resulting curriculum. They can be studied on different age levels, they appear in various areas of computer science and they are relevant also outside of it. We conclude with a short comment on the pilot run evaluation.

Klíčové slová

Výuka informatiky, tvorba kurikula, didaktická transformace, fundamentální ideje, gymnázia.

Keywords

Computer science education, curriculum development, fundamental ideas, grammar schools.

1. ÚVOD

V České republice informatika dosud není zařazena do všeobecného vzdělávání. Žáci se v souladu s Rámcovými

vzdělávacími programy (RVP) učí využívat digitální technologie a pracovat s informačními zdroji, prakticky vůbec se ale nesetkají s informatikou jako vědou. Obvyklá zdůvodnění takového stavu se opírají mimo jiné o názory, že to není potřeba a že to nejde. Teprve ve světle vývoje v zahraničí se česká vláda v roce 2014 rozhodla situaci změnit a informatiku vyučovat.

Abychom si odpověděli na otázku, jestli je výuka informatiky ve všeobecném vzdělávání možná a prospěšná, vytvořili a ověřili jsme program takového úvodu do informatiky pro gymnázium. V rozhodujících parametrech je srovnatelný s dalšími školními předměty jako matematikou či fyzikou. Je také určený všem žákům, nejedná se tedy o učivo pro výběrové semináře.

V tomto článku představíme proces, jakým jsme hledali vhodné pojetí informatiky, které by splnilo výše uvedené požadavky, a výsledek, ke kterému jsme došli. Práci jsme zahájili rozsáhlou rešerší, zahrnující jak vymezení oboru informatiky, tak různá vymezení informatiky jako školního předmětu. Pro naše potřeby jsme se rozhodli využít konceptu fundamentálních idejí. Po jejich identifikaci jsme přistoupili k jejich rozpracování na konkrétní vzdělávací cíle a dále k přípravě programu výuky. Nakonec se podařilo experimentální výuku informatiky realizovat i s úplnou třídou po dobu celého školního roku.

Na začátku článku popíšeme obor, který jsme identifikovali jako výchozí akademickou disciplínu pro experimentální školní předmět informatika. V následující kapitole se věnujeme postupu didaktické transformace oboru na školní předmět a souvisejícím aspektům. V kapitole 4 představíme výsledek transformace a v kapitole 5 výsledky uskutečněné pilotáže.

2. INFORMATIKA JAKO OBOR

Před dalším postupem bylo třeba si ujasnit, co má vlastně být předobrazem budoucího školního předmětu informatika. V češtině se k pojmu informatika váží různé významy, v dalších zemích je situace obdobná. V běžném životě tato mnohoznačnost příliš nevadí. V různých odborných kruzích mnohoznačnost eliminuje ustálený konsenzus o významu pojmů. Za takové situace ovšem nelze beze všeho přistoupit k tvorbě kurikula. Hrubou představu o výuce informatiky jako vědy (v protikladu k uživatelským či technickým dovednostem) bylo nutno značně zpřesnit a ujasnit, co do oboru patří, co je na hranici a co do něj nepatří.

Za tímto účelem jsme prozkoumali množství zdrojů jak českých, tak i zahraničních. Základní vymezení poskytují encyklopedie, zejména při srovnání významu hesel v různých jazycích. Významným zdrojem porozumění je pohled do historie pojmů, ale i samotných oborů. Konkrétnější obsah (a různá pojetí i totožně pojmenovaných oborů) představují sylaby vysokoškolských vzdělávacích programů. Nejpodrobnější pohled potom nabízí

přehledové monografie jako [1–3]. Průzkum nějakým způsobem souvisejících disciplín vymezení informatiky dále zpřesní. Jedná se např. o matematiku, informační technologie, softwarové inženýrství, informační vědu, kybernetiku, lingvistiku a mnohé další obory, které mají s informatikou nějaké styčné plochy.

Provedená rešerše se ukázala jako opodstatněná, na obor informatika existuje množství odlišných pohledů. Na druhé straně se také podle očekávání potvrdilo, že existuje určité společné jádro, ze kterého je možné vycházet. Kdybychom měli zvolit stručné vymezení oboru informatika, patrně bychom volili mezi formulacemi Philipa Guo „Efficiently Implementing Automated Abstractions“ [4], Jeanette Wing „Computing is Automation of Abstraction“ [5] a Petera J. Denninga „the study of information processes, natural and artificial“ [6]. Bližší prozkoumání ukazuje, že se všechna tři vymezení téměř bezezbytku překrývají. Rozdíl je tedy spíš v důrazu na jednotlivé aspekty, např. Guo poněkud zdůrazňuje aplikační stránku informatiky, naproti tomu Denningova formulace se zdá na člověku téměř nezávislá.

Jako velmi užitečná se ukázala související práce Petera J. Denninga [7], kterou zde krátce přiblížíme. Denning se pokusil informatiku kodifikovat. Tradiční struktura vysokoškolského kurikula postavená na jednotlivých technologiích se stávala nepřehlednou pro potřeby popisu informatiky jako oboru. Denning proto v průběhu let identifikoval základní principy, které technologie přesahují, a přitom jsou pro informatiku charakteristické. Výsledek se podobá přístupu v dalších vědách, které se také organizují podle nějakých obecnějších principů. V případě CS se (podle Denninga) jedná o:

- výpočet (computation),
- komunikaci,
- koordinaci,
- automatizaci,
- vybavování (z paměti, recollection).

Uvedené principy se samozřejmě leckde překrývají. Denning v této souvislosti hovoří o jednotlivých oknech, jimiž na informatiku nahlížíme. Uvedený model se v čase vyvíjí. K uvedeným mechanismům např. přibýlo pět kritérií kvalitního návrhu (jednoduchost, výkonnost, spolehlivost, přizpůsobitelnost a bezpečnost). Z nich potom plynou principy jako abstrakce, hierarchizace a další.

Navazující oblast Denningovy práce je snaha o klasifikaci informatiky. Na informatiku lze nahlížet jako na abstraktní vědu (podobnou matematice, s technologiemi coby pouhými aplikacemi), jako na technický obor, jehož úkolem je vytváření technologií a naopak abstraktní teorie jsou druhořadé, nebo jako na přírodní vědu. Pro třetí možnost argumentuje Denning [6, 7]. Ukazuje, jak informatika splňuje standardní požadavky na vědní disciplíny. Informatika nabízí organizovanou poznatkovou základnu, reprodukovatelné výsledky, využívá dobře popsane metody (vč. metod experimentálních), její teorie jsou vyvratitelné a mají prediktivní sílu (a nejsou triviální), zkoumá umělé, ale i přírodní jevy. Nutno podotknout, že většina informatiky a informatiků se zabývá umělými systémy. Myšlenka na přirozené systémy se zdá zprvu podivná. Denning nicméně uvádí přesvědčivé příklady (částicové jevy, zpracování DNA, lingvistika) a další odkazy.

Jedno z využití Denningových principů je tvorba vzdělávacích programů [8]. Výhodou je srozumitelnost (ve srovnání s kurikulem založeným na jednotlivých technologiích), vyváženější obtížnost, vyváženější poměr teoretických konceptů a

praxe a v neposlední řadě přesnější obraz informatiky na veřejnosti.

3. DIDAKTICKÁ TRANSFORMACE

Po prozkoumání výchozího oboru bylo možné započít s jeho transformací do školního předmětu. Zde popíšeme související okolnosti a samotný postup. Školní předměty (resp. vzdělávací obory v RVP) a akademické disciplíny nejsou totožné. Vztahy mezi nimi jsou předmětem zkoumání oborových didaktik [9]. Mezi základní otázky patří výběr vzdělávacích cílů a učiva a jejich zdůvodnění. Proces didaktické transformace lze rozdělit do dvou rovin [10]. Ontodidaktická transformace spočívá v převodu oboru do kurikula, např. ve formě RVP. Psychodidaktická transformace je o úroveň níž, převádí obsah kurikula na skutečnou výuku. Většinou ji tedy realizuje učitel. Školská informatika čelí ve srovnání s dalšími předměty značné neukotvenosti, neexistuje sdílená zkušenost ani názor na běžné a vhodné učivo a jeho uchopení ve výuce. Tento stav lze ale pojmut i jako výhodu, poskytuje totiž příležitost využít zkušeností z ostatních oborů.

Náš výzkum byl koncipován jako design-based research a plány výuky jsme na základě získaných zkušeností opakovaně upravovali. Úpravy se samozřejmě projevovaly na úrovni podrobného plánu, někdy ovšem i na úrovni kurikula. Vždy se nicméně jednalo o dílčí změny. Naopak jsme často hledali, jak dané cíle přeformulovat a dané učivo pojmut jinak, aby je žáci lépe zvládali, a přitom bylo v principu dosaženo původních vzdělávacích cílů. Přes následné úpravy má tedy smysl popsat i prvotní proces transformace.

RVP a realita gymnázií

Zcela zásadním hlediskem při vytváření pojetí informatiky pro gymnázia je samozřejmě situace na gymnáziích samotných. Výuka na nich vychází na jedné straně z Rámcového vzdělávacího programu pro gymnázia (RVP, [11]), na druhé straně z léty zažitých praxe. RVP předepisuje 4 hodiny pro vzdělávací oblast nazvanou Informatika a informační a komunikační technologie. Mnohé školy některé z předepsaných hodin tzv. integrují do výuky jiných předmětů. Oblast je zaměřená na práci s digitálními technologiemi a informačními zdroji. Jediné zřetelně informatické téma je algoritmizace, která představuje jednu položku ze 14 okruhů učiva, resp. jeden z 10 očekávaných výstupů. Čteme-li RVP kreativně, různé části informatiky jako oboru můžeme najít. V takovém případě lze ale jen těžko hovořit o nějakém konceptním pojetí.

Jednou z hlavních změn v souvislosti se zaváděním RVP jsou tzv. klíčové kompetence. Jsou v definovány jako „soubor vědomostí, dovedností, schopností, postojů a hodnot, které jsou důležité pro osobní rozvoj jedince, jeho aktivní zapojení do společnosti a budoucí uplatnění v životě.“ V gymnaziálním vzdělávání rozlišujeme kompetenci k řešení problémů, kompetenci komunikativní, kompetenci k učení, kompetenci sociální a personální, kompetenci občanskou a kompetenci k podnikavosti. Jejich bližší popis je uveden v [12]. Především popis klíčové kompetence k řešení problémů naznačuje, že informatika má v gymnaziálním vzdělání své místo, přestože ji obsahová stránka RVP prakticky úplně pominula.

Vzděláváním na čtyřletých gymnáziích a na vyšším stupni víceletých gymnázií se usiluje o naplnění těchto cílů [11, str. 8]:

- vybavit žáky klíčovými kompetencemi na úrovni, kterou předpokládá RVP G;

- vybavit žáky širokým vzdělanostním základem na úrovni, kterou popisuje RVP G;
- připravit žáky k celoživotnímu učení, profesnímu, občanskému i osobnímu uplatnění.

Přes snahu vymezit vzdělávací obor, který RVP v různých ohledech přesahuje, je namístě udržet přiměřenou shodu s těmito cíli a s celkovou koncepcí RVP. Cílem je tedy najít takové pojetí školské informatiky, které jednak umožňuje rozvíjet klíčové kompetence, dále na přiměřené úrovni seznamuje žáky s informatikou jako svébytným oborem a zároveň jako oborem, který zvyšuje možnosti uplatnění žáků (i mimo informatiku jako takovou). Takový přístup odpovídá již tradičním gymnaziálním předmětům, které sledují obdobné cíle.

Analýza RVP dále poslouží k lepší provázanosti informatiky s ostatní výukou na gymnáziu. Nemělo by např. smysl jen kvůli přidání informatiky nějakou látku zdvojovat. Budeme raději hledat to, co je informatice vlastní, a na gymnáziu chybí, jako např. snaha o efektivitu řešení problémů.

Současný stav výuky na gymnáziích lze využít, zaměřit se čistě na informatiku jako obor a předpokládat, že si žáci potřebné uživatelské dovednosti osvojí už dříve v rámci tradiční výuky, která se s informatikou téměř nepřekrývá.

Celý náš projekt samozřejmě směřuje k tomu, aby na gymnáziích výuka informatiky skutečně probíhala. Takový cíl je ovšem v dosti příkrém rozporu se skutečnostmi jako obsah RVP a nedostatek vyučovacích hodin, které by bylo možno informatice věnovat. Snažíme se přitom o důstojné seznámení s oborem, nikoliv několikahodinový úvod. Přestože některé skutečnosti nezbude než ignorovat (informatika není v RVP), někde se kompromisu nevyhneme (rozsah výuky). V opačném případě by sice vznikla představa výuky informatiky ideální, ovšem zcela nerealistická.

Kurikulum jsme proto počítali pro výuku v jednom školním roce po dvě hodiny týdně. To je sice málo ve srovnání např. s přírodovědnými předměty, ale nepoměrně více než nic. Navíc bylo naším záměrem kurikulum konstruovat modulárním způsobem. Tedy tak, aby bylo možné použít byť jen jeho část, pokud podmínky na dané škole víc nedovolí. Při transformaci jsme nehleděli na to, jestli bude výuka třídy probíhat po polovinách, nebo jestli budou obě hodiny v rozvrhu přímo navazovat (být lze takovou organizaci předpokládat).

Mnohá témata informatiky vyžadují nějaké předchozí znalosti, nejčastěji z matematiky. Předpokládali jsme proto výuku v posledních dvou ročnících gymnázia.

Další aspekty

Cennými zdroji pro transformaci oboru na školní předmět jsou již existující kurikula. Studovali jsme jak starší programy (např. izraelský [13] či slovenský [14]), tak i programy novější, např. ve Spojeném království [15] a USA [16].

Uvedená kurikula nicméně nelze přímo převést do podmínek českých gymnázií. V některém zásadním parametru totiž neodpovídají našim cílům či podmínkám. Některá předpokládají příliš vysokou časovou dotaci (resp. soustavnou výuku v období několika let školní docházky), některá jsou určena jiným věkovým skupinám, další se soustředí především na výuku programování nebo přípravu budoucích studentů informatiky. Cizí kurikula pochopitelně nejsou koncepčně provázána s našimi gymnaziálními předměty.

Nezastupitelný význam při tvorbě informatického kurikula má známý koncept informatického myšlení. Do značné míry totiž odpovídá právě na otázku, co má z informatiky vědět a umět každý člověk. Konkrétních pojetí informatického myšlení existuje větší množství, obvykle vychází z tradice v daném regionu nebo z potřeb místního kurikula a nelze je tedy přímo převzít do našich podmínek. Podrobnější informace a analýzu pojmu informatické myšlení jsme čerpali z článku [17].

Proces didaktické transformace ovlivňuje i obtížnost vznikajícího kurikula. Obtížnost pro žáky zvyšuje už novost předmětu, protože v případě obtíží mají méně možností pomoci než u zavedených předmětů. Informatika má navíc pověst náročného oboru (ovšem v kontextu přípravy na budoucí povolání!). Nakonec poukaz na obtížnost je jedna z častých námitek proti myšlence výuky informatiky ve všeobecném vzdělávání. Chceme-li sestavit kurikulum pro všechny, musíme na otázku obtížnosti brát zřetel.

Lze předpokládat, že hlavním důvodem obtížnosti informatiky bude abstraktnost učiva a nezvyklost řešených problémů. Na rozdíl od nezvyklosti, abstraktnost může být (a v několika případech byla) důvodem k nezařazení nějaké oblasti informatiky do výsledného kurikula. Zkušenost s výukou abstraktních informatických témat nebyla k dispozici, jako přibližné vodítko nám proto posloužilo intuitivní srovnání s abstraktní (či dokonce související), ale již prověřenou látkou ostatních předmětů. Právě srovnatelnost informatiky s ostatními předměty je nakonec jedním z cílů celého snažení. Další heuristikou pro rozhodování o obtížnosti byl předpoklad, že bude snazší případně později ubírat než přidávat, a ověřené pravidlo, že vysoká očekávání vedou k lepším výsledkům.

Dalším aspektem je otázka šířky a hloubky učiva. V různých předmětech se lze setkat s názory, že výuka nemůže jít do potřebné hloubky (nebo dokonce dosahovat uspokojivé kvality), protože množství předepsaného učiva je příliš vysoké. Čistý štít dává šanci v zájmu kvalitního osvojení nezahrnout obsahu příliš mnoho. Snažili jsme se proto omezit na skutečně nejdůležitější základy oboru.

Teoretický rámec didaktické transformace

Teoretická informatika je disciplína velmi bohatá (zvláště s ohledem na hodinovou dotaci na střední škole). Lze ale najít pojmy, které se v různých oblastech informatiky znovu a znovu vracejí, což ukazuje na jejich důležitost. Mohli bychom je přirovnat k množině či indukci v matematice, popř. zákonům zachování ve fyzice. Tyto pojmy mají zásadní význam pro danou vědu, ale také význam, který ji dalece přesahuje. Toto pozorování rozpracoval psycholog Jerome Bruner do podoby *fundamentálních idejí* [18]. Zejména pro oblast výuky informatiky koncept dále rozvinul Andreas Schwill [19].

Fundamentální idea má splňovat několik kritérií. První je tzv. horizontální kritérium, idea má být rozeznatelná na rozličných místech v rámci dané oblasti. Příkladem je stromová struktura (se stromy se setkáme v souborových systémech a různých hierarchiích, v datových strukturách, v modelování alternativních průběhů výpočtů v teorii vyčíslitelnosti, v XML souborech...).

Vertikální kritérium vyžaduje možnost vyučovat ideu na nejrůznějších úrovních složitosti. Algoritmus jako posloupnost kroků je přístupný i nejmladším žákům, vyšší úroveň pak umožňuje zkoumání a porovnání abstraktních výpočetních modelů a hledání obecné definice algoritmu. Pojem NP-úplnosti se naproti tomu mladším žákům zpřístupňuje jen obtížně.

Fundamentální idea musí být rozeznatelná a významná v průběhu delšího historického vývoje dané disciplíny. Toto kritérium má sloužit k rozlišení dlouhodobě významných idejí. To je zvláště citlivá otázka v informatice, která je mezi ostatními poměrně mladou disciplínou. Nadto se velmi rychle vyvíjí, jak poukazují M. Černochová a J. Vaníček [20]. Časem prověřená je např. dvojková soustava, naopak unáhlit bychom se mohli v otázce hesel „big data“ nebo „augmented reality“.

Poslední kritérium vyžaduje, aby idea souvisela se všednodenním jazykem a uvažováním. Příkladem může být každodenně použitelné modelování závislých úkolů pomocí orientovaného grafu. Naopak dvojková soustava sice stojí v základu informatiky, ale (přímo sama o sobě) v běžném životě příliš nepomůže.

Koncept fundamentálních idejí je v různých variantách nadále využíván k rozvoji výuky informatiky. A. Schwill identifikoval jako fundamentální ideje informatiky algoritmizaci, jazyk a strukturovaný rozklad [19]. Z těchto tří potom odvozuje další. Z konceptu fundamentálních idejí ve své experimentální výuce informatiky vychází také Arno Pasternak [21]. V jeho *braided teaching* odpovídá fundamentální idejí pojem *strand* (protože se v průběhu výuky cyklicky vrací, v souladu s Brunerovým *spirálovým kurikulem*), na který jsou ovšem kladeny poněkud jiné požadavky. Hlavní rozdíl spočívá v požadavku, aby *strand* dokázali bez potíží rozpoznat samotní žáci. Kurikulum konstruované tímto alternativním způsobem více reflektuje pohled žáků a jejich vnímání oboru.

V naší práci jsme se rozhodli strukturovat cíle a učivo pomocí fundamentálních idejí. S ohledem na výše popsané aspekty jsme volně sledovali postup publikovaný v [19]. První krok spočívá v analýze obsahu výchozího oboru, jak jsme popsali v předchozí části. Vyhledávání vztahů a podobností mezi různými oblastmi oboru odhaluje první kandidáty na fundamentální ideje. Dále se využijí ostatní kritéria a kandidátské ideje jsou upravovány či vylučovány podle toho, jestli jsou přístupné na různých intelektuálních úrovních, dostatečně historicky podložené a prověřené a relevantní v běžném životě. V závěru práce je zkoumán vzájemný vztah mezi jednotlivými nalezenými idejemi: Jsou na obdobné úrovni abstrakce? Lze je nějak seskupit, uspořádat či jinak strukturovat?

Výsledek se stal základem našeho vymezení školní informatiky. Jednotlivé ideje jsme dále rozkládali, popisovali a přiřazovali k nim související vzdělávací cíle a učivo. Při tom se ještě výrazněji projeví výše zmíněné aspekty, jako předpokládaný rozsah výuky a požadavek srovnatelnosti s dalšími předměty. V této fázi jsme se opírali o monografii [22].

4. INFORMATIKA JAKO ŠKOLNÍ PŘEDMĚT

Nakonec jsme identifikovali čtyři fundamentální ideje, na kterých jsme školní informatiku postavili. Jedná se o informaci, model, algoritmus a efektivitu. Všechny tyto pojmy lze mj. rozpoznat v rámci obecného postupu řešení problému: získáme potřebné informace, vytvoříme vhodný model problému a následně hledáme efektivní, tedy pokud možno algoritmické řešení. Jednotlivé pojmy nyní popíšeme podrobněji.

Informace

Informace jsou materiálem, který v informatice zpracováváme. Zároveň jde o ústřední pojem na úrovni hmotnosti a energie

ve fyzice. Pro žáky jsou relevantní zejména následující dva pohledy na pojem informace. Prvním je informace jako úbytek možností (něco nevím, musím tedy uvažovat několik možností; získáním informace se něco dozvím a některé možnosti vyloučím). Zobecněním tohoto přístupu je klasická informace jako úbytek entropie, což je ovšem pro běžného žáka gymnázia těžko dostupný pojem.

Informace jako úbytek možností je oproti úbytku entropie jednoduchý koncept. Je ovšem dostatečně mocný např. pro odvození algoritmu půlení intervalu. Úzce souvisí s dvojkovou soustavou a tím i druhým významným pojetím informace, totiž dat spolu s jejich interpretací. To vede mj. na problematiku kódování a jazyků.

Žáci by se měli naučit porovnávat informační obsah zpráv a pokládat efektivní otázky, posuzovat efektivitu různých způsobů kódování a provádět základní související výpočty (např. kolik ještě telefonů můžeme připojit do sítě, než začneme přidělovat desetimístná telefonní čísla). Porozumí problematice komprese a skutečnosti, že bezeztrátová komprese má své limity.

Na poněkud abstraktnější úrovni je poznatek, že každou informaci lze zakódovat pomocí posloupnosti znaků a že v případě počítačů k tomu využíváme dvojkovou soustavu.

Model

Žáci samozřejmě mentální modely běžně využívají v ostatních předmětech. Nikde se ale modelům nevěnují explicitně. Pro informatiku je ovšem otázka modelování a jeho kvality stěžejní.

Žáci si rozšíří paletu modelů, které mohou v případě potřeby použít. Především se jedná o různé druhy grafů. Ty sice ve škole i v životě běžně používají (větný rozbor, taxonomie v biologii, vazby v molekulách), ale jen na intuitivní úrovni a neuvědomují si jejich souvislost.

Kromě samotných grafů a jejich typických aplikací se žáci seznámí se způsoby jejich kódování pomocí znaků (tedy nikoliv nakreslení).

Žáci postupně zjistí, že na nakreslení grafů nezáleží, tedy že strukturálně stejný graf (a model téže skutečnosti) může vypadat různě. To vede k jednomu z abstraktnějších témat, k isomorfismu grafů. I v dalších oblastech informatiky se opakovaně vynořuje otázka, jestli dvě struktury nejsou z nějakého hlediska totožné, jen např. jinak označené. V matematice žáci řeší rovnost čísel a geometrickou shodnost. Informatika umožňuje pojem rovnosti dále rozpracovat, např. otázkou na „rovnost“ dvou grafů či algoritmů apod.

Uvedené dovednosti zužitkují při práci s dalším užitečným modelem, kterým je stavový prostor. Žáci jsou sice v různých předmětech vedeni k systematickému řešení problémů, nemají k tomu ale žádný obecný nástroj. Jedním z takových nástrojů je být právě stavový prostor, který je možné systematicky procházet a hledat řešení spolehlivě, bez ohledu na talent či štěstí.

Žáci by se měli naučit rozlišovat různé úrovně abstrakce ve stejné situaci a to, jak jsou vhodné. S tím souvisí vědomí kladů i záporů modelování. K zodpovědnému použití každého nástroje patří znalost jeho limitů. V případě stavových prostorů se jedná především o jejich velikost.

Algoritmus

Algoritmus je ve výuce často vnímán jako „to, co se má naprogramovat“. Takové pojetí ale dostatečně neodpovídá

potřebám gymnazisty. Souvislost s programováním samozřejmě nelze pominout. Algoritmus je ale především spolehlivý postup, který lze realizovat bez dalšího přemýšlení. V principu nezáleží na tom, jestli ho provádí stroj, nebo člověk. Lze předpokládat, že většina gymnazistů bude pracovat právě s postupy „pro lidi“. I ty ovšem často musí být spolehlivé. Typickým příkladem, kde se v principu pokoušíme splnit požadavky na algoritmy, jsou oblast medicíny a práva.

Žáci se seznámí s různými způsoby zápisu algoritmů. V životě se nejčastěji setkají s postupy popsány přírodním jazykem, jimž proto věnujeme zvláštní pozornost. Dále pracujeme s vývojovými diagramy (a rozhodovacími stromy jako jejich předstupněm).

Žáci by se měli setkat i se základy programování. Cílem je nicméně především podpora porozumění informatickým principům, nikoliv tvorba softwaru. Žáci se stručně seznámí se základními řídicími strukturami a s tím, jak jim programování může ušetřit práci. Pro většinu z nich je nakonec hlavním výsledkem právě schopnost rozpoznat rutinní úlohu, kterou nemusí řešit člověk, a zjištění, že mohou leckdy sami drobnou úpravou změnit chování různých programů.

Konkrétní algoritmy se žáci běžně učí a používají je v ostatních předmětech. Obvykle ovšem nedojdou až k jejich přesnému zápisu. Stejně tak nerozlišují, jestli jsou předkládané postupy algoritmické (typicky v matematice, fyzice) nebo nikoliv (např. v jazycích).

Žáci by měli umět (na přiměřené úrovni obtížnosti) rozpoznat, jestli je daný postup algoritmem a případně ho vhodně upravit, aby se algoritmem stal. Měli by umět algoritmus opravit nebo přizpůsobit algoritmus jinému účelu.

Žáci by měli umět algoritmická řešení i vytvářet. K tomu by měli být vybaveni řadou příkladů, kterými se mohou inspirovat a z nichž mohou vycházet. Mohou se naučit algoritmy jako vyhledávání půlením intervalu, řazení nebo systematické procházení grafu (stavového prostoru). Přitom nečekáme, že je v praxi použijí přímo a beze změny z paměti. Měli si ale osvojit a použít příslušné základní principy (např. „rozděl a panuj“).

Pod algoritmus řadíme i rekurzi jako elegantní způsob řešení některých problémů. Těžko očekávat, že budou žáci sami rekurzivní řešení objevovat. Měli by si s nimi ale poradit, rekurze (ať už algoritmická, nebo strukturální) by je ale neměla zmást.

Tak jako v ostatních případech, i u algoritmů by žáci měli vědět, že mají své limity a v základních případech usoudit, kdy je nutno vyzkoušet jiný přístup k řešení. Jedná se jak o problémy s neurčitým zadáním, tak o problémy neřešitelné v principu, protože řešení vůbec neexistuje, nebo protože nelze algoritmicky najít.

Efektivita

Otázka efektivity se prolíná takřka celou informatikou. Tradiční součástí (vysokoškolských) úvodních kurzů je problematika asymptotické složitosti. To je pro gymnazisty příliš formální, mohou se nicméně naučit řádově odhadovat prostředky potřebné pro vykonání toho kterého postupu v příznivém, průměrném i nepříznivém případě. Tím pádem by měli umět dva algoritmy porovnat a zvolit ten vhodnější. Zároveň by měli umět pracovat i s dalšími kritérii, nikoliv jen s v teorii oblíbeným počtem základních operací jako zástupcem měření času.

Efektivita staví další limit použitelnosti algoritmických řešení. Pojem NP-úplnosti je pro gymnázium příliš abstraktní a málo

potřebný. I gymnazista by měl nicméně vědět, že exponenciální algoritmus nejspíš použít nechce.

Pod efektivitu řadíme i rozličné heuristiky k jejímu zvyšování, tedy optimalizaci. Jako příklad uveďme předtřídění dat nebo okamžité ukončení práce ve chvíli, kdy máme dostatečný výsledek. Sem zapadá i otázka paralelizace. Žáci se seznámí s modelováním závislostí navazujících úkolů pomocí orientovaného grafu a se základními možnostmi použití, vč. zjišťování mezní doby splnění všech úkolů a počtu souběžně prováděných úkolů.

Už v souvislosti s algoritmem žáci zjišťují, na jaký typ úkolů se algoritmické postupy hodí, a na jaké se naopak hodí člověk. V konečném důsledku se jedná o otázku efektivní alokace zdrojů. Opět se vynořuje otázka pojetí rovnosti: pokud člověk i stroj odvedou stejně kvalitní práci, bude stroj nejspíš levnější. Žáci se proto musí seznámit také se stavem a potenciálem vývoje v oblasti umělé inteligence.

Program výuky

Ve výuce není nutně nejvhodnější postupovat po jednotlivých idejích. Někteří učitelé by např. pojem algoritmus stavěli co možná nejvíc na začátek výuky, právě s poukazem na fundamentálnost pojmu algoritmu pro informatiku. Nám se nicméně osvědčilo posečkat, až se žáci přirozeně seznámí s několika příklady. Díky nim si pak mohou definiční vlastnosti algoritmů představit konkrétně, lépe jim tak porozumět, a lépe porozumět také tomu, proč jsou právě ony tak významné.

Jednotlivé fundamentální ideje se navíc překrývají v tom smyslu, že na totéž učivo lze pohlížet různě a zdůraznit jeho jinou stránku. Například rozhodovací stromy jsou *modelem* rozhodování, tedy nějakého *algoritmu*, který vede k získání *informace*. Snažíme se přitom, aby stromy nebyly zbytečně hluboké, tedy aby rozhodování probíhalo *efektivně*. S podobnou situací se setkáme u grafových notací. Představují způsob zakódování nějaké *informace*, která sama něco *modeluje*. Přitom bychom rádi, aby způsob práce s tou kterou notací byl *algoritmický* a abychom vybrali notaci v nějakém smyslu *efektivní* – např. jednoduše použitelnou, nebo datově úspornou.

Program výuky jsme tedy strukturovali do jednotlivých volně navazujících modulů, každý věnovaný jednomu hlavnímu tématu. Moduly ovšem nejsou izolované, základní pojmy se opakovaně vynořují a prolínají se všemi moduly, podobně jako v programu [21]. Pojmenování modulů je tak spíš pro orientaci a pro zdůraznění důležitosti jednotlivých témat z pohledů žáků. Jeden modul je plánován přibližně na jeden měsíc výuky.

Podrobné informace lze získat v on-line učebnici, kterou na základě získaných zkušeností tvoříme [23].

5. PILOTÁŽ

Pro úplnost článku je na místě doplnit, že uvedené výsledky nejsou čistě teoretické. Navržené pojetí informatiky jsme nejprve několik let testovali a upravovali v rámci různých volitelných seminářů, abychom jej nakonec jeden školní rok pilotovali ve výuce s celou třídou žáků.

Dostali jsme k dispozici kvintu osmiletého gymnázia, ve které jsme mohli vyučovat povinnou informatiku podle námi vyvinutého pojetí. Žáci měli jednu hodinu týdně, výuka probíhala po polovinách. Ze 32 žáků bylo 17 děvčat.

Oproti plánu se jednalo o mladší žáky a menší rozsah výuky, což pochopitelně způsobilo jisté komplikace. Na druhé straně takové nastavení umožnilo ještě přísněji určit skutečně základní vzdělávací cíle a formulovat je tak, aby nezávisely na úrovni znalostí z ostatních předmětů. Ukázalo se, že je to možné, a nevede to nutně k trivializaci látky.

Výuku jsme hodnotili na základě různých souborů dat, zahrnujících naše vlastní poznámky, práce žáků, data ze školního informačního systému a výsledky pololetních a závěrečných dotazníků, které žáci vyplňovali. Závěry z pololetních dotazníků jsme publikovali v [24], celkové závěry k publikaci připravujeme.

Sami žáci deklarovali přínos výuky ve schopnosti efektivního řešení problémů i efektivního chování obecně, v abstraktním (či logickém) myšlení a ve schopnosti jednoznačného a srozumitelného vyjádření. Tyto výsledky potvrzuje i srovnání úrovně žákovských prací na začátku a na konci školního roku. Lze tedy tvrdit, že jsme výukou pozitivně ovlivnili klíčové kompetence žáků.

Poněkud překvapivě uváděli žáci i zlepšení v oblasti uživatelských dovedností. Na to jsme se ve výuce přímo nezaměřovali. Postupovali jsme nicméně tak, jak to očekáváme i v jiných předmětech: když bylo pro výuku něco potřeba, naučili jsme se to. Důležitým faktorem je patrně také to, že množství vzdělávacích aktivit probíhalo na počítačích.

V parametrech jako zajímavost, užitečnost a oblíbenost (vše z pohledu žáků) se informatika bez problémů řadila mezi přírodní vědy a matematiku. Jednoznačně nejvýše v porovnání s ostatními předměty žáci hodnotili obtížnost pochopení informatiky. Tento výsledek byl kromě obtížnosti samotné látky nutně ovlivněn i věkem žáků, existencí domácích úkolů a jejich nezvyklou povahou (např. nebylo vždy hned zřejmé, jak postupovat) a použitím systému Moodle. Zajímavé je, že se obtížnost rozhodujícím způsobem neprojevila např. na oblíbenosti.

Další oblast zkoumání se týkala pojetí informatiky jako takové. Zatímco na začátku roku viděli žáci předmět téměř výlučně jako ovládání aplikací, po roce výuky se v jejich vlastních definicích objevilo 22krát slovo věda, 22krát slovo technologie a opakovaně se vyskytovaly vazby jako „co je uvnitř“ či „jak to funguje“. Představa těchto žáků o informatice se tedy zřetelně přiblížila skutečnosti.

6. ZÁVĚR

Na uvedené výsledky je samozřejmě třeba pohlížet jako na výsledky v dané třídě, v daném roce a na dané škole. Postačí ale jako ukázka toho, že výuka vědní informatiky podle výsledného programu je možná a může být pro žáky prospěšná. Zároveň je nutno dodat, že rozsah testování a pilotáže stěžil mohl vést k optimálnímu nastavení vzdělávacích cílů a metod. Je tedy nadále na čem pracovat a co zlepšovat.

7. BIBLIOGRAFICKÉ ODKAZY

- [1] VANÍČEK, J., M. PAPÍK, R. PERGL and T. VANÍČEK. *Teoretické základy informatiky*. Praha: Kernberg Publishing, 2007.
- [2] AHO, AV and JD ULLMAN. *Foundations of computer science*. New York: Computer Science Press, 2000.
- [3] HAREL, David and Yishai FELDMAN. *Algorithmics: The spirit of Computing*. Addison-Wesley: Addison-Wesley, 2004.
- [4] GUO, Philip J. *What is Computer Science? Efficiently Implementing Automated Abstractions* [online]. 2010. Dostupné z: <http://pgbovine.net/what-is-computer-science.htm>
- [5] WING, Jeannette M. *Computational thinking*. B.m.: Carnegie Mellon University. 2011.
- [6] DENNING, Peter J. Computing is a natural science. *Communications of the ACM*. 2007, vol. 50, no. 7, pp. 13–18.
- [7] DENNING, Peter J. Is computer science science? *Communications of the ACM*. 2005, vol. 48, no. 4, pp. 27–31.
- [8] DENNING, Peter J. Great Principles in Computing Curricula. In: *SIGCSE '04 Proceedings of the 35th SIGCSE technical symposium on Computer science education*. ACM New York, NY, USA, 2004, p. 336–341.
- [9] DENG, Z. Transforming the subject matter of a secondary-school science subject. *Journal of Curriculum Studies*. 2007, vol. 39, no. 5, pp. 503–535.
- [10] JANÍK, Tomáš and Jan SLAVÍK. Vztah obor - vyučovací předmět jako metodologický problém. *Orbis Scholae*. 2007, vol. 1, no. 1, pp. 54–66.
- [11] *Rámcový vzdělávací program pro gymnázia*. Praha: Výzkumný ústav pedagogický v Praze, 2007. ISBN 978-80-87000-11-3.
- [12] ŠLEJŠKOVÁ, E. *Klíčové kompetence na gymnáziu*. Praha: Výzkumný ústav pedagogický v Praze, 2008. ISBN 9788087000205.
- [13] GAL-EZER, Judith, Catriel BEERI, David HAREL and Amiram YEHUDAI. A High-School Program in Computer Science. *Computer*. 1995, vol. 28, no. 10, pp. 73–80.
- [14] *Štátny vzdelávací program pre gymnázia v Slovenskej republike ISCED 3A – Vyššie sekundárne vzdelávanie*. Bratislava: Štátny pedagogický ústav, 2008.
- [15] COMPUTING AT SCHOOL WORKING GROUP. *Computer Science: A curriculum for schools*. Computing at School, 2012.
- [16] TUCKER, Allen B., Deborah SEEHORN, Stephen CAREY, Daniel MOIX, Brian FUSCHETTO, Irene LEE, Dianne O'GRADY-CUNIFF, Chris STEPHENSON and Anita VERNÓ. *CSTA K-12 Computer Science Standards*. 2011.
- [17] LESSNER, Daniel. Analýza významu pojmu „Computational Thinking“. *Journal of Technology and Information Education*. 2014, vol. 6, no. 1, pp. 71–88.
- [18] BRUNER, Jerome Seymour. *The process of education*. B.m.: Harvard University Press, 1977.
- [19] SCHWILL, Andreas. Fundamental Ideas of Computer Science. *Bull. European Assoc. for Theoretical Computer Science*. 1994, vol. 53.
- [20] VANÍČEK, Jiří and Miroslava ČERNOCHOVÁ. Didaktika informatiky na startu. In: STUHLÍKOVÁ and JANÍK, eds. *Oborové didaktiky v České republice: stav, vývoj a perspektivy*. Brno: Masarykova Univerzita, 2015, in press.
- [21] PASTERNAK, Arno and Jan VAHRENHOLD. Design and evaluation of a braided teaching course in sixth grade computer science education. In: *Proceedings of the 43rd ACM technical symposium on Computer Science Education*. New York, NY, USA: ACM, 2012, p. 45–50.
- [22] PASCH, Marvin, Trevor G. GARDNER, Georgea M. LANGER, Alane J. STARK and Christella D. MOODY. *Od vzdělávacího programu k vyučovací hodině*. Praha: Portál, 2005. ISBN 80-7367-054-2.
- [23] LESSNER, Daniel. *Informatika pro každého* [online]. 2015. Dostupné z: <http://ksvi.mff.cuni.cz/ucebnice>
- [24] LESSNER, Daniel. Jak žáci gymnázia vnímají výuku informatiky jako vědy. In: *Didinfo 2013 - Zborník príspevkov*. Banská Bystrica: Univerzita Mateja Bela, 2013.