

HOW DO WE TRANSLATE COMPUTATIONAL THINKING INTO CZECH?

Daniel Lessner

Abstract: The article introduces the concept of computational thinking (CT). This ability “to think like a computer scientist while solving problems” draws more and more attention in relation to computer science education. We show and discuss a few definitions of CT. The concept can then be compared to key competences and literacies as used in Czechia.

The second half of the article illustrates the theoretical specifications with a variety of examples from both everyday life and school education. Our “computer science” classes focus on ICT user skills. CT provides a convenient starting point for the quest for what is missing in our education.

Key words: computer science education, computational thinking.

JAK SI PŘELOŽÍME „COMPUTATIONAL THINKING“?

Resumé: Článek představuje koncept „computational thinking“ (CT). Tato schopnost „myslet jako informatik při řešení problémů“ si v souvislosti s výukou informatiky získává stále více pozornosti. V článku proto stručně probereme a porovnáme několik odlišných vymezení CT, abychom mohli koncept následně srovnat s kategoriemi českému čtenáři bližšími, totiž s klíčovými kompetencemi a gramotnostmi.

Ve druhé polovině článku uvedená teoretická vymezení ilustrujeme množstvím příkladů použití CT v životě i ve výuce. Koncept CT poskytuje vhodné východisko pro hledání toho, co v naší výuce „informatiky“ zaměřené na používání technologií chybí.

Klíčová slova: computational thinking, informatické myšlení, didaktika informatiky.

1 Úvod

Není nutno rozvádět, jak významně mění tvář světa rychle se vyvíjející informatika. Ne každému je ovšem zřejmé, že změny nepřicházejí jen díky technickému pokroku. Ve světě proto nabývá na zvučnosti koncept *computational thinking* (CT).

Pro začátek nám postačí představa CT jako schopnosti „myslet jako informatik při řešení problémů“. Takto pojaté CT má být užitečné i mimo informatiku a snad i univerzálně, tedy pro každého.

Balením školní brašny žák projevuje informatické myšlení, předběžně načítá (nebo spíš nakládá) do pracovního úložiště. Maminka myslí jako informatik, když zorganizuje jogurty v chladničce do prioritní fronty podle data spotřeby. Výběr nejrychlejší fronty v supermarketu vyžaduje modelování výkonnosti. Telefonní linka fungující i při výpadku proudu ukazuje redundanci a nezávislost selhání v návrhu. V jednoduchých situacích postačí „selský rozum“. Dále v článku popíšeme i složitější situace, v nichž se už uplatní schopnost použít znalosti z informatiky.

CT je pokusem uchopit právě „to ono“, co při řešení (i neinformatických) problémů odlišuje

informatiky a neinformatiky, a co se často intuitivně snažíme při výuce v žácích probudit a rozvíjet.

Takový koncept je v souladu s převažujícím vývojem názorů na informatiku v primárním a sekundárním vzdělávání ve světě. Výuka informatiky tak už nesměruje jen ke zvládnutí užívání ICT pro běžný život a případná výuka programování neslouží primárně k ovládnutí syntaxe jazyka a tvorbě programů. Cílem je obdoba kultivace myšlení, ke které má docházet např. ve výuce matematiky (vedle např. tréninku početních dovedností).

V češtině je ovšem o CT minimum dostupných informací. V tomto článku jej proto blíže představíme českým zájemcům.

Na stručný historický úvod navážeme ukázkou a srovnáním několika nejdůležitějších vymezení CT od různých autorů. Dále stručně prozkoumáme vztah CT s klíčovými kompetencemi a gramotnostmi užívanými v českém prostředí¹. Tuto teoretickou diskusi

1 Někteří čtenáři by možná čekali srovnání CT s Rámcovými vzdělávacími programy jako takovými, tedy včetně vzdělávacích oblastí, průřezových témat atd. Souvislosti s CT bychom přirozeně hledali především v oblastech

doplníme ukázkami konkrétních situací a náměty použitelnými ve výuce.

Smyslem příspěvku je seznámit čtenáře s konceptem CT a prozkoumat, jak může CT prospět výuce informatiky na našich školách. Přirozeně totiž vyvstává otázka, jestli je CT jen jiný úhel pohledu na to, co už máme obsaženo např. právě v klíčových kompetencích, nebo jestli přináší i něco nového a má smysl se jím dále zabývat – a snad najít i vhodný český překlad.

2 Původ CT

První použití sousloví computational thinking je připisováno Seymouru Papertovi, autorovi původní verze programovacího jazyka LOGO, určeného ke konstruktivistickému vzdělávání [1, 2], a jedné z vůdčích postav iniciativy One Laptop per Child. V článku [3] Papert diskutuje, jaké změny přináší možnost využití počítačů do výuky matematiky, pojem CT nicméně dále nerozvíjí.

Současný zájem o CT spustila Jeanette Wing z Carnegie Mellon University (CMU) až článkem [4]. Představila v něm vizi CT jako schopnosti podobně základní, jako je čtení, psaní a počítání, a především podobně potřebné. Wing se primárně nesnaží o rozšíření řad uchazečů o studium informatiky. Snaží se ukázat, že porozumění informatice se hodí jak profesionálům napříč obory, tak při řešení běžných každodenních problémů – obojí pochopitelně na odpovídající úrovni složitosti.

Článek původně nebyl určen vzdělavatelům, ti si nicméně všimli, že by CT mohlo pomoci lépe uchopit vzdělávací cíle a aktivity v souvislosti s informatikou. Pojmenovává totiž důležitou (a obtížně uchopitelnou) stránku výuky: nikoliv použití technologií, nikoliv dovednosti IT profesionálů, nýbrž způsob přemýšlení, který si díky informatice můžeme osvojit.

Wing do CT řadí širokou škálu myšlenkových nástrojů. Na jedné straně stojí samozřejmě rozložení problému nebo jeho převedení na jiný. Na druhé straně potom např. i vyhýbání se uváznutí (deadlocku) v distribuovaných systémech, resp. skutečných situacích, které jimi lze modelovat.

matematika a informatika. I rychlý pohled ale ukáže, že není co zkoumat. Odhlédneme-li od klíčových kompetencí, RVP ZV rozvoj CT nezahrnuje. To není žádným překvapením, když si uvědomíme, že nezahrnuje ani výuku informatiky. Pro srovnání doporučujeme slovenský Štátny vzdelávací program.

Uplatnění CT má podle Wing umožnit lepší řešení starých (již nějak vyřešených) problémů a také řešení problémů, kterým by jinak bylo nutno se vyhnout, nebo by na ně jako na problémy k vyřešení vůbec nebylo nahlíženo.

3 Vymezení CT

Definovat CT je podobně záludné, jako definovat klíčové kompetence či gramotnosti. Výsledky jsou často buď neurčité a obtížně použitelné pro samotnou výuku, nebo sice dostatečně konkrétní, ale tím pádem dlouhé a pravděpodobně příliš úzké. V této části uvedeme několik takových pokusů o vymezení CT. Dosud totiž nebyla nalezena shoda na tom, jak CT definovat a strukturovat.

Jako první se podívejme na definici **Jeanette Wing**. V původním vizionářském a tedy poměrně neurčitém článku žádnou nedává, místo ní popisuje následující vlastnosti. CT je dovednost základní, tedy nutná pro plnohodnotné fungování v moderní společnosti. Jde o způsob myšlení lidí, nikoliv strojů, CT není mechanické. CT kombinuje a doplňuje matematické a technické myšlení. Používané modely jsou matematické (jako v každé vědě), jsou ale omezené konstrukčními možnostmi strojů. Z druhé strany, informatika je podobná technice, protože produkuje nástroje interagující s fyzickým světem. Vytváří si ovšem také světy vlastní, fyzikou neomezené. Už v původním článku autorka uvedla pozoruhodný rys CT, který se v různých obměnách objevuje i v dalších definicích: CT samozřejmě zahrnuje konceptualizaci, vyžaduje ovšem uvažování *na několika úrovních abstrakce zároveň*.

Až později jsme se dočkali definice od autorky v [5, 6]:

CT jsou myšlenkové postupy zapojené při takovém formulování problémů a jejich řešení, které umožní tato řešení efektivně provést agentem zpracovávajícím informace.

Zmíněným agentem může být jak stroj, tak člověk. CT obecně má zahrnovat schopnosti

- pochopit, které aspekty problému jsou řešitelné stroje;
- vyhodnotit shodu mezi informatickými prostředky a problémem;
- porozumět možnostem a omezením informatických prostředků;
- použít informatické prostředky novým způsobem či v nové situaci (nebo prostředky přizpůsobit);

- použít informatické strategie v jakékoliv oblasti.

Informatickým prostředkem se zde rozumí jak technické nástroje, tak abstraktní postupy a teoretické výsledky.

Uvedený popis se vztahuje na každého, autorka dále uvádí rozšíření pro vědce, techniky a další profesionály. To se týká především řešení problémů s pomocí velkých dat a použití informatické terminologie. Připomeňme, že definice J. Wing má fungovat obecně, není určena do školství jako vzdělávací cíl. Určení obsahu a dovedností vhodných pro různé stupně vzdělávání je třeba hledat jinde.

Jedna z konkrétnějších a velmi používaných definic pochází ze spolupráce **International Society for Technology in Education (ISTE)** a **Computer Science Teachers Association (CSTA)** [7]. Téměř totožná definice se objevuje v [8].

CT je postup řešení problému, který zahrnuje mimo jiné následující charakteristiky:

- Formulovat problémy způsobem, který umožňuje jejich strojové řešení
- Logicky uspořádat a zkoumat data
- Reprezentovat data prostřednictvím abstrakcí, jako jsou modely a simulace
- Automatizovat řešení pomocí algoritmického myšlení (jako posloupnost kroků)
- Odhalit, prozkoumat a provést možná řešení s cílem odhalit nejúčinnější kombinaci činností a zdrojů
- Zobecňovat a přenášet tento postup řešení problémů do nejrůznějších dalších oblastí

Tyto dovednosti jsou podpořeny předpoklady a postoji, které jsou taktéž nezbytnou součástí CT:

- Sebejistota tváří v tvář složitosti
- Vytrvalost při řešení obtížného problému
- Snášení nejednoznačnosti
- Schopnost vypořádat se s otevřenými problémy
- Schopnost dorozumět se a spolupracovat s ostatními při dosahování společného cíle

Oblíbenost této definice vyplývá z její struktury. Charakteristika CT je zde dostatečně konkrétní k tomu, aby umožňovala plánování výukových aktivit. Mezi ostatními vyniká tato definice také zahrnutím postojové části. Ta mimochodem dále zpřesňuje, o jaké problémy se v souvislosti s CT jedná, což je opět velmi

užitečné při plánování výuky. Navíc si můžeme všimnout, že uvedené postoje u nás cíleně téměř nerozvíjíme.

Britská **Královská společnost** formulovala svůj vlastní přístup:

CT je postup rozpoznávání informatických aspektů světa kolem nás a využití informatických prostředků k porozumění a uvažování o přirozených i umělých systémech a procesech [9].

Definice je na velmi vysoké úrovni, přitom je ale poměrně srozumitelná. Neopírá se příliš o termíny z informatiky, naopak zdůrazňuje roli CT v životě člověka a roli informatiky jako přírodní vědy pro zkoumání světa. Ve svojí obecnosti je tedy v jistém protikladu k ostatním definicím. V pozadí lze spatřit podobnou myšlenkovou konstrukci, s jakou se setkáváme v definici *matematické gramotnosti*, tedy (zkráceně) dostatečnému porozumění matematice spolu se schopností jej využít v běžném životě (přesněji viz dále).

Následující vymezení nejsou přímo zaštitěna vědeckou prací, právě proto ale vhodně dokreslují celkový obrázek. Vlastní web o CT ve vzdělávání provozuje společnost **Google** [10]. Kromě množství příkladů uvádí i svou definici:

CT zahrnuje sadu technik a dovedností k řešení problémů, které při psaní běžně používaných aplikací (vyhledávání, email, mapy), používají softwaroví inženýři. CT je nicméně využitelné téměř v jakémkoliv předmětu. Součástí CT jsou zejména

- rozklad problému,
- rozpoznávání vzorů (např. v grafech na burze, ale i v procesech),
- zobecňování vzorů (tedy vytváření abstraktních modelů),
- navrhování algoritmů.

Toto pojetí CT je nepochybně přizemnější. Zdá se, že čerpá především z „řemesla“ kodérů a „vědu“ informatiků zcela pomíjí. Patrně ale není na místě v tom hledat záměr „nemáme v Googlu dost programátorů“. Je nutno mít na paměti poněkud posunuté anglické významy „*computer science*“ a „*computing*“ vůči české informatice. Google na svých stránkách uvádí značné množství příkladů výukových aktivit, které jsou programování velmi vzdálené (viz níže). Navíc uvedené čtyři základní složky nejsou nikterak triviální. Tvoří přitom logický obecný rámec řešení problémů.

Další odlišností uvedeného vymezení CT je výslovné zahrnutí rozpoznávání vzorů jako jedné z hlavních složek CT. To do konceptu velmi

přirozeně zapadá a vede ke vzniku mnoha neobvyklých a zajímavých výukových aktivit.

Posledním kamínkem mozaiky vymezení pochází z webu **Computer Science For Fun** [11], který provozuje Queen Mary University of London. CT je podle jeho autorů *sada rozličných dovedností, které souvisí s řešením problémů a vyplývají ze zkoumání povahy zpracování informace. Zahrnuje jak dovednosti rozvíjené většinou předmětů (jako tvořivost, schopnost vysvětlování a týmové práce), tak i několik velmi specifických dovedností řešení problémů, jako schopnost logického, algoritmického a rekurzivního myšlení.*

Informatika má tyto rozličné dovednosti jedinečným způsobem spojovat. CT potom autoři rozkládají na další druhy myšlení: logické, algoritmické, efektivní, vědecké, a inovativní. Celkově ale toto vymezení trpí právě odkazy na další koncepty, které již ale nejsou definované.

Na závěr této části vymezení krátce srovnáme. Co se týče rozdílů, jsme svědky různé úrovně konkrétnosti, různě silného sepětí CT a programování či CT a teoretické informatiky.

Hledání společných znaků pak odhalí především zaměření na řešení široké škály problémů, použití abstrakce (často dokonce na několika úrovních zároveň), důraz na znalost a tvůrčí použití principů informatiky namísto znalostí uživatelských, přesahy do dalších oblastí (logika, vyjadřování, práce v týmu, ostatní přírodní vědy...) a snahu o efektivitu výsledných řešení (vč. posouzení vhodnosti strojového řešení a jeho mezí). To vše má být přitom návazné i na problémy každodenního života, a CT tak má být součástí všeobecného vzdělání.

Snad je tedy nyní konkrétněji patrné, co je myšleno spojením „myslet jako informatik“. Šíře uvedených charakteristik CT by již měla stačit k tomu, aby si o CT čtenář udělal poměrně jasnou představu. Kromě toho si může vybrat, které pojetí nejlépe odpovídá potřebám jeho žáků a výuky (popřípadě různé přístupy zkombinovat).

4 Klíčové kompetence a CT

Čeští učitelé se v současnosti potýkají s dlouhou řadou kompetencí, gramotností a dalších poměrně abstraktních konceptů. Je proto na místě vyjasnit, v jakém vztahu k nim je CT. Přirozeně totiž vyvstává otázka, jestli není CT jen jiný úhel pohledu na to, co už máme obsaženo např. právě v klíčových kompetencích, tudíž to ve výuce rozvíjíme a koncept CT nepotřebujeme.

Výkladu pojmu klíčové kompetence najdeme v odborné literatuře více [12], vzhledem k účelu

článku jsme se rozhodli přidržet výkladu z Rámcového vzdělávacího programu pro základní vzdělávání² a souvisejících dokumentů.

Jako první se přirozeně zaměříme na *kompetenci k řešení problémů* (KŘP). Podle očekávání najdeme s CT značný překryv. Popis KŘP využívá slova a formulace jako promyšlený systematický postup, rozhodování, kritické myšlení, předvídání neočekávaných situací, práce s informacemi, diagnostika chyb, překonávání neúspěchu a dokonce i zobecňování výsledných řešení.

Přitom není informatika nikde zmíněna výslovně. To je poněkud škoda s ohledem na to, že matematika a logika výslovně uvedeny jsou. Přitom je to právě informatika, která poskytuje nástroje pro mnohé součásti KŘP, vč. výše uvedených.

Vztah KŘP a informatiky je podobný vztahu KŘP a matematiky. Zvládnutí informatiky či matematiky jistě není postačující podmínkou vysoké úrovně KŘP. Je ale jednoznačně podmínkou nutnou. Čteme-li specifikaci KŘP a rozumíme při tom možnostem využití informatiky pro řešení problémů, je souvislost zcela zřejmá. Je však poněkud zarážející, že popis nepoužívá přilehavou informatickou terminologii. Důsledkem je, že většina uživatelů popisu KŘP souvislost s CT neuvidí, není to totiž zvykem.

Je zároveň nutno dodat, že KŘP ve skutečnosti CT důsledně nepokrývá. V popisu KŘP nenalezneme nástroje jako modely či abstrakce (což je samo o sobě zvláštní), ani náznaky preference strojového, čili algoritmického řešení. Podobně chybí jasné hodnocení efektivity různých řešení.

Tyto skutečnosti samozřejmě nic nemění na tom, že KŘP je patrně nejsilnějším pojítkem

2 Nutnost výběru je dána rozdíly a rozpory mezi jednotlivými dokumenty. RVP G nahrazuje kompetenci pracovní z RVP ZV kompetencí k podnikavosti. RVP S se pak liší ještě výrazněji, když mezi klíčové kompetence řadí i kompetence využívat prostředky informačních a komunikačních technologií a pracovat s informacemi a kompetence matematické. To je podstatné s ohledem na to, že CT je na rozdíl od obecných klíčových kompetencí na základní škole oborové, navázané na obsah informatických předmětů. V dalším se proto omezíme na klíčové kompetence základní školy, které se týkají největšího počtu žáků.

CT a RVP a že výuka informatiky pojaté jako rozvoj CT může významně posilovat KŘP.

Celkem logicky a očekávaně nalezneme souvislosti CT s kompetencemi k učení a pracovní. V příslušných popisech uváděné plánování, provádění a hodnocení učebních i pracovních postupů přímo souvisí s algoritmickou stránkou CT.

Neměli bychom ovšem přehlédnout kompetenci komunikativní (KK). Komunikace je ve velmi úzkém vztahu s pojmem informace, na kterém informatika (a tím i CT) stojí. Práce s informací je koneckonců součástí KK (byť bez ohledu na otázky kódování či efektivity).

Součástí CT je i schopnost používat rozličné formální jazyky, ať už pro přesnou komunikaci s lidmi, nebo pro komunikaci přímo se stroji. Tento zásadní aspekt ovšem KK zcela pomíjí (kromě dále nerozpracované zmínky o symbolických prostředcích).

KK je celá vystavěna kolem mezilidské komunikace přirozeným jazykem. Dokonce i část věnovaná (uživatelskému) použití ICT předpokládá především komunikaci mezi lidmi. Přitom mnoho zákonitostí funguje v obou oblastech stejně právě proto, že jde o výměnu informací, bez ohledu na to, jestli některá strana komunikace žije.

Celkově lze konstatovat, že klíčové kompetence pokrývají pouze základy CT. Na druhé straně se CT jeví jako vhodný prostředek k jejich rozvoji. Je v pořádku, že se specifikace jednotlivých kompetencí na informatiku (případně CT) přímo neodkazují, jsou koneckonců jako nad-oborové zamýšleny. Je proto ovšem třeba věnovat zvláštní pozornost při interpretaci popisů, abychom na příslušné součásti CT nezapomínali.

5 Gramotnosti a CT

Podívejme se dále krátce na vztah CT a některých gramotností. RVP s nimi sice přímo nepracují, přesto se nimi ve školství setkáváme [13]. Jako nejbližší CT se může jevit *ICT gramotnost*. Ta se ovšem zaměřuje především na použití ICT. Sice do ní spadá i zpracování informací, ale zdaleka ne tak pokročilé, jaké očekáváme v CT. Zjednodušeně lze říci, že se CT a ICT gramotnost překrývají do té míry, do jaké je třeba pro CT skutečně využít techniku.

Mnohem zajímavější (a bližší) je vztah CT k *matematické gramotnosti*. Připomeňme definici [13]:

Matematická gramotnost je schopnost jedince poznat a pochopit roli, kterou hraje matematika

ve světě, dělat dobře podložené úsudky a proniknout do matematiky tak, aby splňovala jeho životní potřeby jako tvořivého, zainteresovaného a přemýšlivého občana. Záměnou několika slov se můžeme dostat překvapivě blízko CT. Totéž platí i pro podrobný popis složek matematické gramotnosti.

Na místě je proto otázka, kde leží hranice matematické gramotnosti a CT. Přeložíme-li si doslovně anglický výraz computational, nejde o velké překvapení. Hranice je tedy daná prakticky tím, kde chápeme hranici mezi (školskou) matematikou a informatikou. V kontextu vzdělávání lze zjednodušeně říci, že se matematika soustředí spíše na zadání a správný výsledek, kdežto informatika se soustředí spíše na samotný proces hledání toho výsledku a jeho zobecnění pro celou třídu problémů. Opět zjednodušeně, v matematice algoritmy používáme, zatímco v informatice je hledáme a zkoumáme.

Odtud se pak obě vědy rozcházejí a začínají řešit odlišné druhy otázek. Jak už bylo uvedeno, informatika se omezuje na fyzicky realizovatelné a automatizovatelné modely. Navíc je žádoucí, aby realizace modelů byla efektivní.

Uvedené rozdíly snadno nahlédneme např. z obsahu učebnic, z typických úloh a dalších učebních aktivit a z hodnocených dovedností.

CT se tedy odlišuje zaměřením (efektivita a algoritmčnost řešení) a tím pádem někdy pokročilejším a abstraktnějším obsahem. Základní cíl, totiž úspěšné řešení problémů, ale s matematickou gramotností sdílí. Vzhledem k již existujícím výsledkům na poli výzkumu matematické gramotnosti by bylo zajímavé vyzkoušet podle jejího vzoru strukturovat CT (situace a kontexty, kompetence, obsah).

Za zmínku stojí také vztah CT a *přírodovědné gramotnosti*, vzhledem k tomu, že informatika je přírodní věda. Opět se opakuje situace, že informatika, pokud čteme pozorně, vlastně zahrnuta je, ovšem nikoliv výslovně. Překryv tedy najdeme do té míry, do jaké informatika (a tím i CT) používá metody přírodních věd uvedené v popisu přírodovědné gramotnosti.

Informatika (stejně jako ostatní přírodní vědy) pak používá i postupy v popisu z nějakého důvodu neuvedené, především simulaci a modelování.

6 Příklady ze skutečnosti

Příklady použití CT v běžném životě mohou být zcela jednoduché. Učitel tance na konci lekce vyzývá k přihlášení zvednutím ruky ty páry,

kterým vyhovuje páteční termín závěrečného plesu, a spočte zvednuté ruce. Následně se ptá, komu vyhovuje termín sobotní. Ke zvednutí ruky ale již vyzývá jen dámy. Učitel tance právě na okamžik zauvažoval jako informatik a uvědomil si, jak zkrátit čas počítání přibližně polovinu.

Okamžitě se nabízí dvě námitky. První: není to matematika? Ne. Matematicky jsou správně oba postupy, oba dávají správný výsledek. Při výuce matematiky se zkoumáním efektivity vyučovaných postupů zabýváme velmi zřídka.

Námítka druhá: učitel tance k tomuhle nemusí být informatik, stačí mu přece zdravý rozum. Ano. Tak jako v případě ostatních předmětů, i informatické problémy lze, pokud jsou jednoduché, správně vyřešit intuitivně. Odpověď leží v kvalitě života, které chceme dosahovat. Jistě lze šťastně žít i bez CT. V principu podobně, jako lze šťastně žít bez dovednosti číst, psát a počítat.

Ukažme si nyní příklad složitější [5]. Trumpetista a informatik Roger Dannenberg si měl stejně jako ostatní členové kapely vybrat daných asi 40 skladeb pro daný večer z neseřazené složky asi dvou set. Všichni ostatní začali postupně procházet složku a po jednom hledat a vybírat skladby. R. Dannenberg se rozhodl seřadit 200 skladeb v čase $O(N \cdot \log(N))$ a teprve potom najít hledané skladby, tedy už v čase $O(M \cdot \log(N))$, namísto $O(M \cdot N)$, tak jako ostatní (N je velikost složky, zde 200, M je počet hledaných skladeb, tedy 40). Sice pořad ještě řadil, zatímco ostatní byli zpola hotovi a podívovali se jeho počínání, skončil nicméně jako první.

Poznamenejme, že v takové situaci nestačí použít heuristiku „vždycky je lepší je řadit“ nebo horní odhady složitosti. Pokud totiž nejsou vstupy přesvědčivě velké nebo nemáme dostatečnou zkušenost, není ihned zřejmé, jestli řazení není zbytečná práce. Kdyby byl rozdíl v počtu skladeb k vyhledání a počtu skladeb ve složce výraznější, mohlo by být rychlejší přímo hledat. Dále je nutno si uvědomit, že $O(M \cdot N)$ zde nelze ztotožnit s $M \cdot N$, pravděpodobně bychom totiž náš odhad o 100% nadhodnotili. Trvání úkonů při řazení a při hledání není jednoduše zaměnitelné. Další otázkou je vhodná volba řadícího algoritmu pro dané podmínky (tedy řazení nikoliv čísel v paměti, ale papírů v lidských rukách). Pochopitelně platí, že čím větší problém je, tím

častěji převáží (správně použitá) obecná informatická teorie nad konkrétními okolnostmi.

A nakonec si položíme otázku, jestli by nebylo bývalo lepší celou úlohu obrátit naruby. Postupně projít složku a pro každou skladbu (kterou stejně vezmeme do ruky a přečteme její název) zkontrolovat, jestli se se nachází v předem seřazeném seznamu 40 vybraných skladeb. Celkově potřebujeme čas $O(M \cdot \log(M))$ na seřazení seznamu a $O(N \cdot \log(M))$ na vyhledání vybraných skladeb, tedy řešení v tomto případě ještě lepší. Navíc je toto řešení obecně lepší častěji, tedy ve více obdobných situacích.

Tyto úvahy měly ukázat, že k efektivnímu vyřešení problému nepostačí pouhé znalosti informatiky. Je nutno jako informatik přemýšlet, znalosti použít i v nových situacích. Zahrnutí tradičního obsahu vědní informatiky do výuky proto samo o sobě CT nutně nerozvíjí. Výpočet očekávané složitosti Quick Sortu není rozhodující (byť užitečný) v situaci, kdy je hlavním omezením efektivity fakt, že máme k dispozici pouze dvě ruce.

Podívejme se nyní na odbornější příklady využití CT. Článek [14] uvádí množství prací, které využívají prorůstání biologie a informatiky. Pěknou ukázkou je práce [15]. Dobrovolníci zasílají nahrávky zpěvu strnadů z různých míst republiky. Nahrávky jsou pak analyzovány a zkoumá se, jak se na různých místech liší nářečí strnadů. Bez digitální techniky by byl takový sběr dat nemyslitelný. Tisíce nahrávek jsou ovšem, navzdory dnešním možnostem, klasifikovány ručně.

Na závěr této části uvedeme vážnější příklad [16]. Je jedním z výsledků výzkumu CT na CMU. V USA každoročně smrtelně onemocní 30 000 lidí, kterým by pomohla transplantace ledvin. Často se v rodině najde ochotný, ale nekompatibilní dárc. Proto má smysl hledat vzájemné kombinace dárců a příjemců tak, aby bylo zachráněno maximum životů. Řešení, které není nejlepší, nebo které je nalezeno pozdě, znamená, že zemře někdo, kdo zemřít nemusel.

Zapojení CT na straně organizátorů dárcovského systému vedlo k rozpoznání problému a posouzení jeho řešitelnosti. Následná spolupráce s informatiky vedla ke kvalitativnímu zlepšení situace. To přitom nespočívá v nějakém technickém zařízení. Spočívá v chytřejší práci s dostupnými informacemi.

7 CT ve výuce informatiky

Následující příklady už více či méně souvisí se školní výukou. Zařazení CT do výuky

informatických předmětů je poměrně přímočaré. Výuku je třeba rozšířit o pokročilejší informatická témata (na odpovídající úrovni obtížnosti), jako jsou efektivita a omezení informatiky.

Výše jsme již ovšem ukázali, že samotné zařazení příslušných témat nestačí. Je třeba zauvažovat o tom, nakolik zahrnutá témata k rozvoji CT skutečně přispívají. Jiným příkladem může být výuka „programování“: od výuky algoritmizace jako metody řešení problémů a způsobu nahlížení na svět je v praxi velmi snadné sklouznout k výuce programování jako zapisování programů v programovacím jazyce. Do diskusí o prvním programovacím jazyce se pak vkrádají nesouvisející argumenty, jako kolik procent programátorů daný jazyk využívá v práci. Koncept CT nám dává možnost si takový posun uvědomit a napravit.

Může se tak ukázat, že pokud je naším cílem seznámit žáky s ideou omezené množiny příkazů, rozhodování, opakování, rekurzivního volání či ladění, postačí jednoduchý robotanik [17]. S nejmladšími žáky lze pracovat např. v prostředí šachové Figurkové školičky [18]. Odlišný typ zkušenosti nabízí výuka s pomocí robotů.

Prvotní inspiraci mimo programování jako takové lze hledat např. v archivu úloh dostupném na webu Bobříka informatiky. Úlohy z bobříka ukazují, jak mohou informatické principy pomoci k efektivnímu řešení problémů i mimo tradičně vnímanou oblast informatiky. Příznačné jsou pak některé reakce učitelů i soutěžících, které způsob přemýšlení nutný pro řešení úloh vůbec nespojují s informatikou [19].

Ucelenějším materiálem pro výuku je celosvětově úspěšná sada aktivit Computer Science Unplugged [20], která zábavným a zároveň odborně správným způsobem představuje informatiku i nejmladším dětem. Skutečnost, že předmětem zkoumání informatiky nejsou počítače, ujasňují aktivity CSU jednoduše tím, že počítače nevyužívají. Novozélandský tým stojí také za učebnicí [21], která je určena pro práci se staršími žáky. Její přínos spočívá ve zpřístupnění mnoha partií informatiky, které např. v Česku někdy považujeme za příliš náročné.

8 CT ve výuce mimo informatiku

Pro náš článek je zajímavější zapojení CT v neinformatických předmětech. Cílem přitom není udělat z žáků informatiky. Začneme několika doporučeními.

Prvním krokem k zařazení CT je položit si otázku: Jak lze probírané problémy efektivně řešit v dnešních podmínkách? Tradiční metody práce (hledání v papírovém slovníku, ruční řešení soustavy rovnic) samozřejmě neztrácí svůj didaktický význam. Žáci se ale musí seznámit i s tím, jak se daná činnost řeší ve skutečnosti, včetně toho, jaké další problémy z daného oboru lze díky informatice řešit (např. cílené navrhování makromolekul s žádanými vlastnostmi). Nejčastější použití CT mimo informatické předměty souvisí s pokročilou analýzou rozsáhlých dat, modelováním a simulacemi. Simulace jsou koneckonců vedle experimentů a teorie považovány za třetí pilíř vědy [22]. Další informace k zapojování CT do výuky nalezneme v [23].

Rozvoj CT v neinformatických předmětech není na jejich úkor, naopak. Informatický pohled na dané téma žákům umožní hlubší pochopení souvislostí. Např. v biologii se tak nutně budou ptát, jak přesně spolu souvisí genetická informace v buňce a částečná informace v pohlavní buňce: jak jsou vybrány ty části? Podobně, od rodičů sice dědíme geny, přitom nám ale předávají celé chromozomy, kterých je o mnoho řádů méně. Jak je to tedy s výslednou rozmanitostí? Různé úrovně abstrakce si přeci musí odpovídat. Zapojení CT tedy umožňuje také odhalování miskonceptů. Dalším nástrojem jejich odhalování je samostatná formulace probíraných postupů jako algoritmů (např. hledání kořenů kvadratické rovnice formou vývojového diagramu).

Bez ohledu na předchozí je pak důvodem pro rozvoj CT v neinformatických předmětech fakt, že se CT, tedy aspoň za zdmi škol, stalo jejich přirozenou součástí. Chce-li tedy žák oboru plně porozumět, popř. se jím zabývat ve svém dalším životě, neobejde se bez příslušných souvislostí (např. jazykové korpusy a strojové překlady, geografické informační systémy, simulace a modelování, sběr a vytěžování velkých dat atp.).

Konkrétní příklady uvedeme záměrně méně obvyklé, abychom tak naznačili širší dostupných možností. Začneme v oblasti *Člověk a svět práce*. Rozvíjet CT lze snadno o optimalizaci pracovních postupů. Jak nejlépe zorganizovat obsah ledničky? Jak nejlépe zorganizovat vybavení kuchyně? Co znamená "jak nejlépe"? Jde o ušetřený čas při práci, snadný úklid, co nejvíce ušetřeného prostoru? Každý může mít svoje priority a z nich vyplyne optimální organizace. Přínos spočívá v uvědoměném a promyšleném rozhodování: „přestože jsem od maminky zvyklý jinak, uložím mléko do dveří

ledničky, protože je tam snáz dostupné, takže ušetřím čas.“ U pracovních postupů (recepty, konstrukční návody) můžeme nechat žáky uvažovat nad závislostmi jednotlivých úkonů. Které z nich je nutno dělat v daném pořadí? Které lze dělat zároveň? Kolik kamarádů mi může pomoci? Jak můžeme být nejrychleji hotovi? Na tyto přirozené otázky následně hladce naváže látka informatiky, která umožní žákům získané intuitivní poznatky strukturovat, zobecnit a dát do souvislosti.

Ve výtvarné výchově lze pracovat s fraktálními ornamenty – ať už ručně, nebo s pomocí technologií [24]. Kromě toho, že jsou výsledky samy o sobě esteticky hodnotné, přímá zkušenost žáků s rekurzí jim nenásilně zprostředkuje několik zásadních poznatků. Dá překvapivě hodně práce (resp. počítači dlouho trvá) dostat se byť do malé hloubky. Složitě útvary lze někdy překvapivě jednoduše popsat. Tím pádem ovšem není jednoduše vidět, k jakému výsledku daný popis vede.

V matematice se lze věnovat problémovým úlohám a přípravě na jejich informatické řešení. Zakreslení stavového prostoru úlohy o převozníkovi hlídajícím vlka, kozu a zelí či úloh o přelévání žákům ukáže jiný pohled na řešení problémů. Aniž by bylo nutno hned probrat algoritmy procházení grafů, žáci zjistí, že ač se to zprvu nezdá, lze postupovat systematicky. Talent pak není podmínkou úspěchu při řešení, žáci poznávají, že k němu vede i správně nasměrovaná a zorganizovaná píle.

Dalším netradičním příkladem budiž řešení Grelling-Nelsonova paradoxu [25], který lze pojmut jako jazykové cvičení. Přídavná jména jako „libozvučné“, „konečné“ či „pětislabičné“ můžeme považovat za *samopopisná*, tedy *popisující sama sebe*. Adjektiva „jednoslabičné“ či „anglické“ samopopisná nejsou. Zdá se, že lze o každém adjektivu říci, jestli samopopisné, či nikoliv. Jaké je tedy „*samopopisné*“? Jedná se o self-referenční paradox podobný např. paradoxu lháře. Paradox lháře patří sice na první pohled do logiky a ne do informatiky, ale jen dokud si neuvědomíme, že je informatika na logice postavena.

Kromě toho hraje paradox lháře důležitou úlohu při zkoumání *problému zastavení*³ a tím

3 Self-referenci podle vzoru paradoxu lháře využíváme ke konstrukci stroje, který ukáže, že předpoklad řešitelnosti problému zastavení vede ke sporu.

tedy i hledání odpovědi na otázku, co vypočítat lze, a co už nikoliv. To je klíčové téma informatiky a pochopitelně i CT. Můžeme odhadovat, že žáci, kteří se již nad paradoxem zamýšleli, snáze porozumí i pozdějším úvahám, již přímo souvisejícím s problémem zastavení. Proto uvedený paradox řadíme mezi příklady rozvoje CT.

Podobně důležitý je v informatice fakt, že algoritmy s exponenciální časovou složitostí jsou v praxi často k ničemu. Tomu lze ovšem mnohem snáze porozumět na základě předchozí zkušenosti s exponenciálním vývojem a jeho rychlostí (i mimo informatiku).

Množství dalších námětů pro různé předměty a různé věkové skupiny uvádí [10, 26, 27]. Jako příklady zde zmíníme modelování přepisu DNA na proteiny, zaplňování orbitalů, radioaktivního rozpadu, letu balistických střel, zkoumání (hypotetické či skutečné) černé skříňky, kódování a šifrování, skupinovou tvorbu příběhu s následným hledáním a odstraňováním logických nesrovnalostí, odvození gramatických pravidel (např. z ukázkových dat) a jejich algoritmicke formulaci.

9 Závěr

Jak nám tedy může koncept CT prospět? Ve výuce informatiky nám pomáhá oprostít se od zaměření na konkrétní technologie a jejich užívání. Může posloužit jako rámec pro strukturování a formulování pokročilejších kognitivních vzdělávacích cílů.

Dále CT osvětluje význam informatiky pro ostatní obory, souvislosti předmětů mezi sebou i s životem mimo školu. Dává tak vyučujícím různých předmětů půdorys pro diskuse a spolupráci. Za hlavní důvod pro zapojení CT do jejich výuky (kromě aktualizace látky) lze přitom považovat snahu o prohloubení porozumění probíraným jevům.

Cílem rozvoje CT na školách není výchova populace programátorů. CT je v různé míře užitečné pro každého. Umožňuje řešit problémy, jejichž rozsah a složitost je překážkou řešení jiným způsobem. Navíc právě otevřené a komplexní problémy jsou ty, které nedovedeme řešit přímočaře a mechanicky, a je tedy nutno uplatnit tvůrčí lidskou mysl. Zároveň jsou to právě ty problémy, kterým se ve výuce mnozí žáci (někdy spolu s učiteli) rádi vyhnou.

Považujeme-li za základní dovednost počítání, měli bychom s ohledem na vývoj lidského poznání uvažovat o přijetí CT mezi základní dovednosti. Lze na něj nahlížet jako na rozšíření

počítání ve svém nově objeveném významu, totiž zpracování informací.

CT umožňuje racionální rozhodování i v obtížnějších situacích a pomáhá dopady našich rozhodnutí předvídat. Z násobuje možnosti každého člověka ve světě, ve kterém roste význam informací a kde je výpočetní technika všudypřítomná. Pomáhá přitom technickému vývoji, jeho důsledkům, rychlosti, i principiálním limitům porozumět. To se může ukázat jako zásadní mj. pro správné kariérní rozhodnutí [28].

Schopnost automatizovat řešení problémů člověka v konečném důsledku osvobozuje, umožňuje mu totiž zacházet se svým časem podle svého přání.

Ve světle v článku uvedených informací proto pokládáme otázku: jak si přeložíme computational thinking?

10 Literatura

- [1] PAPERT, Seymour a Idit HAREL. Situating Constructionism. In: *Constructionism*. Ablex Publishing Corporation, 1991, p. 193–206.
- [2] BLAHO, Andrej a Ivan KALAŠ. *Imagine Logo - programování pro děti*. Brno: Computer Press, 2006.
- [3] PAPERT, Seymour. An Exploration in the Space of Mathematics Educations. *International Journal of Computers for Mathematical Learning*. 1996, Springer, vol. 1, no. 1, pp. 95–123.
- [4] WING, Jeannette M. Computational thinking. *Communications of the ACM*. 2006, vol. 49, no. 3, pp. 33–35. ISSN 00010782.
- [5] WING, Jeannette M. *Computational Thinking: What and Why?* [online]. 2010 [cit. 2013-10-28]. Dostupné z: <https://www.cs.cmu.edu/~CompThink/papers/ThelinkWing.pdf>
- [6] CUNY, Jan, Larry SNYDER a Jeannette M. WING. *Demystifying Computational Thinking for Non-Computer Scientists*. 2010
- [7] *Operational Definition of Computational Thinking for K-12 Education* [online]. B.m.: International Society for Technology in Education (ISTE) a Computer Science Teachers Association (CSTA). 2011. Dostupné z: <http://csta.acm.org/Curriculum/sub/CurrFiles/CompThinkingFlyer.pdf>
- [8] STEPHENSON, Chris a Valerie BARR. Defining Computational Thinking for K-12. *CSTA Voice*. 2011, vol. 7, no. 2, pp. 3–4.
- [9] FURBER, Steve. *Shut down or restart? The way forward for computing in UK schools* [online]. London: The Royal Society, 2012. Dostupné z: http://royalsociety.org/uploadedFiles/Royal_Society_Content/education/policy/computing-in-schools/2012-01-12-Computing-in-Schools.pdf
- [10] GOOGLE, Inc. What is CT? *Exploring Computational Thinking* [online]. [cit. 2013-10-28]. Dostupné z: <http://www.google.com/edu/computational-thinking/what-is-ct.html>
- [11] CURZON, Paul, Peter BLACK a Jonathan MCOWAN. What is Computational Thinking? *Computer Science for Fun* [online]. [cit. 2013-10-28]. Dostupné z: <http://www.cs4fn.org/computationalthinking/index.php>
- [12] BELZ, Horst a Marco SIEGRIST. *Klíčové kompetence a jejich rozvíjení*. Praha: Portál, 2001.
- [13] KOLEKTIV. *Gramotnosti ve vzdělávání* [online]. Praha: Výzkumný ústav pedagogický v Praze, 2010. ISBN 9788087000410. Dostupné z: http://www.nuv.cz/uploads/Publikace/vup/Gramotnosti_ve_vzdelavani11.pdf
- [14] NAVLAKHA, Saket a Ziv BAR-JOSEPH. Algorithms in nature: the convergence of systems biology and computational thinking. *Molecular Systems Biology*. 2011, vol. 7, no. 546.
- [15] DIBLÍKOVÁ, Lucie. *Pták roku 2011: zapojení veřejnosti při analýze regionální variability ptačího zpěvu*. B.m., 2013. Univerzita Karlova v Praze.
- [16] ABRAHAM, David J, Avrim BLUM a Tuomas SANDHOLM. Clearing Algorithms for Barter Exchange Markets: Enabling Nationwide Kidney Exchanges. In: *Proceedings of the 8th*

ACM conference on Electronic commerce. New York, NY, USA: ACM, 2007, p. 295–304.

[17] JARUSEK, Petr a Radek PELÁNEK. A web-based problem solving tool for introductory computer science. In: *Proceedings of the 17th ACM annual conference on Innovation and technology in computer science education - ITiCSE '12*. New York, New York, USA: ACM Press, 2012, p. 371. ISBN 9781450312462.

[18] KOŘENOVÁ, Martina. *Figurková školička* [online]. 2006 [cit. 2013-10-28]. Dostupné z: <http://skolicka.figurka.net/>

[19] LESSNER, Daniel a Jiří VANÍČEK. Bobřík učí informatiku. *Matematika – fyzika – informatika* [online]. Praha: Prometheus, 2013, vol. 22, no. 5, pp. 374–382. Dostupné z: <http://mfi.upol.cz/index.php/mfi/article/view/92/105>

[20] BELL, Tim, Ian H WITTEN, Mike FELLOWS, Robyn ADAMS a Jane MCKENZIE. *Computer Science Unplugged: An enrichment and extension programme for primary-aged children* [online]. 2006. Dostupné z: <http://csunplugged.org/>

[21] BELL, Tim a Jack MORGAN. *Computer Science Field Guide* [online]. 2012 [cit. 2013-10-28]. Dostupné z: <http://www.cosc.canterbury.ac.nz/csfieldguide/>

[22] WILSON, Kenneth G. Grand challenges to computational science. *Future Generation Computer Systems*. 1989, vol. 5, no. 2-3, pp. 171–189 [cit. 2013-10-28]. ISSN 0167739X.

[23] LEE, Irene, Fred MARTIN, Jill DENNER, Bob COULTER, Walter ALLAN, Jeri ERICKSON, Joyce MALYN-SMITH a Linda WERNER. Computational thinking for youth in practice. *ACM Inroads*. 2011, vol. 2, no. 1, p. 32. ISSN 21532184.

[24] SCHACHMAN, Toby. Alternative programming interfaces for alternative programmers. In: *Proceedings of the ACM international symposium on New ideas, new paradigms, and reflections on programming and software - Onward! '12*. New York, New York,

USA: ACM Press, 2012, p. 1. ISBN 9781450315623.

[25] GRELLING, Kurt a Leonard NELSON. Bemerkungen zu den Paradoxien von Russell und Burali-Forti. In: *Abhandlungen der Fries'schen Schule II*. 1907, p. 301–334.

[26] DWYER, Hilary, Bryce BOE, Charlotte HILL, Diana FRANKLIN a Danielle HARLOW. Computational Thinking for Physics: Programming Models of Physics Phenomenon in Elementary School. In: *Physics Education Research Conference (PERC) Proceedings* [online]. Portland, 2013. Dostupné z: <http://www.compadre.org/PER/document/ServeFile.cfm?ID=13126&DocID=3675>

[27] PHILLIPS, Pat. *Computational thinking - a problem solving tool for every classroom* [online]. CSTA & Microsoft Corporation, 2009. Dostupné z: <http://csta.acm.org/Resources/sub/ResourceFiles/CompThinking.pdf>

[28] FREY, Carl Benedikt a Michael A OSBORNE. *The future of employment: how susceptible are jobs to computerisation?* [online]. 2013. Dostupné z: http://www.futuretech.ox.ac.uk/sites/futuretech.ox.ac.uk/files/The_Future_of_Employment_OMS_Working_Paper_0.pdf

Mgr. Daniel Lessner
Kabinet software a výuky
informatiky
Matematicko-fyzikální fakulta UK
Malostranské nám. 25
118 00, Praha 1, ČR
E-mail: lessner@ksvi.mff.cuni.cz
Www: ksvi.mff.cuni.cz/~lessner