

# Fyzikálna Eliza

Michal Malý, Miroslav Kocun

Katedra aplikovanej informatiky FMFI UK  
Mlynská dolina, 824 48 Bratislava  
maly@ii.fmph.uniba.sk, miroslav.kocun@st.fmph.uniba.sk

## Abstrakt

Vytvorili sme program, schopný riešiť jednoduché fyzikálne slovné úlohy. Podobne ako Witzenbaumova ELIZA, využíva hľadanie kľúčových slov, jednoduchú štruktúru úloh a predprogramované vzorce. Analyzujeme dosiahnuté výsledky a limity použitého prístupu. Uvádzame tiež možné vylepšenia metódy, ktoré by vylepšili úspešnosť programu.

## 1 Úvod

Keď Witzenbaum napísal svoj program ELIZA [6], bola to malá revolúcia. Na jednej strane ukázal, že napísať program, ktorý by oklamal bežného človeka, nie je až také ťažké. Na druhej strane prevládlo sklamanie, že program tomu, čo hovorí, vlastne vôbec nerozumie – iba využíva niektoré kľúčové slova a predprogramované šablóny na generovanie odpovede. ELIZA ukázala limity a možnosti mechanického „slovíčkárenia“.

## 2 Motivácia

Bežnou súčasťou školského vyučovania fyziky, matematiky, ako aj iných predmetov, je riešenie slovných úloh. To má preukázať, že žiak učivu rozumie, že ovláda nielen sadu naučených vzorcov, ale vie ich aj prakticky uplatniť.

Pri pohľade na slovné úlohy sme si položili otázku, či je možné naprogramovať program, ktorý by uspokojivo riešil slovné úlohy z fyziky<sup>1</sup>. Fyziku sme vybrali preto, lebo spomedzi ostatných predmetov má dobre definované vstupné a výstupné údaje (napr. rozmery fyzikálnych veličín).

Mohí učitelia sa sťažujú, že ich žiaci riešia úlohy mechanicky: nerozumejú im. Napriek tomu prechádzajú školským systémom. Je zaujímavé zistiť, nakoľko sa naozaj dajú úlohy riešiť mechanicky a nakoľko je potrebné ich porozumenie. Naša aplikácia toto demonštruje, a ukazuje tak, prečo je potrebné voliť rôzne typy úloh: nielen výpočtové úlohy, ale aj úlohy, kde je

potrebné podať len vysvetlenie alebo uviesť príklad k danej tematike.

Nositel' nobelovej ceny za fyziku R. P. Feynman zaviedol [7] pojem vedy v štýle kargo kultu (*cargo cult science*) a uvádzal príklady študentov, ktorí sa dokázali memorovať definície a počítať komplikované príklady, ale v skutočnosti úlohám nerozumeli a nedokázali ich použiť v praxi.

## 3 Algoritmus

### 3.1 Východiská

Pri pohľade na bežnú slovnú úlohu zistíme, že má relatívne jednoduchú štruktúru: Uvedenie vstupných údajov a otázka, prípadne príkaz niečo vypočítať. Ako predmet otázky/príkazu býva niektorá fyzikálna veličina.

### 3.1 Metóda

Aktuálna verzia programu používa jednoduchú analýzu textu. Hľadá kľúčové reťazce, ktoré spadajú do štyroch kategórií: fyzikálne veličiny (napr. „rýchlosť“), čísla (napr. „100“), fyzikálne jednotky (napr. „km/h“), a pomocné slová (napr. „je“). Kľúčové slová sa hľadajú bez ohľadu na ich gramatický tvar, čo znamená, že sa berie do úvahy iba prefix slova („rýchl\*“), prípadne sú vymenované tvary („bol“, „je“, „bude“ a pod.). Ako prvé sa hľadajú čísla a priradí sa k nim fyzikálna jednotka. Podľa jednotky sa určí fyzikálna veličina. Ďalej sa analyzuje forma a predmet otázky. Všetky spomenuté údaje sa uložia, pričom jednotky sa premenia na základné. Tým sú dané vstupné a výstupné údaje, a ďalej tiež pomocné údaje ako napr. tvar slovesa v otázke.

Potom sa hľadá aplikovateľný vzorec. Ten sa použije a podľa neho sa vygeneruje postup riešenia príkladu a výsledok. Jednotky sa zvolia vhodne tak, aby odpoveď bola v ľudsky čitateľnej podobe. Slovná odpoveď sa vygeneruje v gramatickom čase podľa uloženého tvaru slovesa v zadaní a použije sa základný gramatický tvar veličiny („Rýchlosť je ...“, „Rýchlosť bola ...“ a pod.)

Technické detaily parsovania sú popísané v [4].

<sup>1</sup> Poďakovanie za inšpiráciu patrí Jánovi Oravcovi.

### 3.1.1 Názov pod-podkapitoly

Text prvej časti vnútornej do tretej podkapitoly.

## 4 Výsledky

Aplikácia je vytvorená formou Java Appletu a je voľne prístupná na [Chyba! Nenašiel sa žiaden zdroj odkazov.]. Rieši jednoduché fyzikálne úlohy z preddefinovaných oblastí, momentálne sú to príklady na rýchlosť, zrýchlenie, silu, tlak, mechanický výkon. Vyskúšali sme zadať niekoľko príkladov zo školských učebníc, ktoré neboli použité počas vývoja aplikácie.

### 4.1 Riešenie úloh

Uvádzame niekoľko vyriešených príkladov – dialógov používateľa (U) a Elizy (E). Eliza poskytuje analýzu príkladu, riešenie a slovnú odpoveď tak, ako sa zvykne vyžadovať pri školských úlohách.

**Príklad 1** Riešenie príkladu na rýchlosť (z [Chyba! Nenašiel sa žiaden zdroj odkazov.]

U: Pri maratónskom behu prebehol bežec úsek trate dĺžky 4 km rovnomerným pohybom za 13 minút. Akou rýchlosťou prebehol tento úsek?

E: Dráha:  $s = 4.0 \text{ km}$  (4000.0 m) Čas:  $t = 13.0 \text{ min}$  (780.0 s)  
 $v = s / t$   
 $v = 4000.0 / 780.0$   
 $v = 5.128 \text{ m/s}$   
Rýchlosť je 18.462 km/h.

**Príklad 2** Riešenie príkladu na zrýchlenie (z [Chyba! Nenašiel sa žiaden zdroj odkazov.]

U: S akým veľkým zrýchlením sa na štartovacej dráhe rozbieha lietadlo, ktoré za 20 sekúnd dosiahne rýchlosť 150 m/s?

E: Čas:  $t = 20.0 \text{ sek}$  (20.0 s) Rýchlosť:  $v = 150.0 \text{ m/s}$  (150.0 m/s)  
 $a = v / t$   
 $a = 150.0 / 20.0$   
 $a = 7.5 \text{ m*s}^{-2}$   
Zrýchlenie bolo 7.5 m\*s<sup>-2</sup>.

**Príklad 3** Riešenie príkladu na tlak (z [Chyba! Nenašiel sa žiaden zdroj odkazov.]

U: Kolmo na voľnú hladinu kvapaliny v nádobe pôsobí piest s obsahom 0,25 m<sup>2</sup> tlakovou silou 56 N. Aký veľký tlak v kvapaline vznikne?

E: Plocha:  $S = 0.25 \text{ m}^2$  (0.25 m<sup>2</sup>) Tlaková sila:  $F_t = 56.0 \text{ N}$  (56.0 N)  
 $p = F_t / S$   
 $p = 56.0 / 0.25$   
 $Tlak = 224.0 \text{ Pa}$   
Tlak je 224.0 Pa

### 4.2 Generovanie úloh

Program okrem riešenia aj generuje úlohy pre zadanú alebo náhodne zvolenú veličinu. Tieto úlohy je možné nakopírovať na vstup a nechať ich hneď programom vyriešiť. Takáto funkcionálna je vhodná napr. pre učiteľov, ktorí potrebujú vygenerovať rôzne úlohy pre rôznych žiakov alebo pre rôzne skupiny. Eliza takto môže slúžiť ako elektronická "nevyčerpatelná" zbierka úloh, pravda, z obmedzenej domény a úlohy sú len obmedzeného počtu typov.

**Príklad 4** Generovanie ľubovoľného príkladu.

U: Eliza, daj mi hocijaký príklad.

E: Nech sa páči:  
Autobus prešiel z bodu A do bodu B za 346.755 min. Priemerná rýchlosť bola 61.498 km/h. Vypočítajte vzdialenosť.

**Príklad 5** Generovanie príkladu pre zadanú veličinu.

U: Eliza, daj mi príklad na výkon.

E: Tu je tvoj príklad:  
Robot v továrni vykonal prácu 421.895 KJ počas doby 648.525 min. Zistite jeho výkon.

## 5 Diskusia

### 5.1 Chyby a obmedzenia programu a možné vylepšenia

Príklady, ktoré program správne nevyrieši, by sa dali rozdeliť do niektorých kategórií podľa toho, aký princíp by bol potrebný na jeho vyriešenie. Sú zoradené podľa náročnosti ich vyriešenia.

- **iná doména** Program nevyrieši príklady z inej oblasti, na ktorú nemá vzorce. Tu je náprava jednoduchá: stačí doprogramovať potrebné vzorce. Prípadne by sa mohol učiť (viď nižšie).
- **odolnosť voči chybám** Program nevyrieši príklady, v ktorých je preklep alebo iná podobná chyba. Na preklepy by sa dala použiť jednoduchá metóda – nájsť najbližšie kľúčové slovo (napr.

„rýchlost“ → „rychlost“, hoci je tu isté riziko, že pri jednotkách by to nemuselo byť jednoznačné. Úplné vynechanie údajov je väčší problém: človek, ktorý problému rozumie, vie povedať, ktorý údaj je potrebný na vyriešenie úlohy a poprípadne si úlohu aj vhodne parametrizovať (pozri nižšie). Toto by sme sčasti vedeli vyriešiť hľadaním vzorcov, ktoré by vyhovovali existujúcim údajom, no pravdepodobne by prichádzalo do úvahy viac možností. Tieto možnosti by sa dali selektovať v procese učenia.

- **úlohy s parametrami** Niektoré úlohy pre pokročilých obsahujú miesto číselných hodnôt premenné. Takéto parametre program nerozpozná. Program by bolo možné rozšíriť o detekciu názvov premenných, ktoré majú rozpoznateľné mená (napr.  $x, y, p, t$ ) ktoré nie sú zameniteľné so slovami. Pre tie premenné, ktoré by sa mohli zameniť s bežnými slovami (napr.  $a, s, z$ ), by bolo potrebné vykonať gramatickú analýzu. Riešenie by prebiehalo najprv v rovine symbolového výpočtu a potom prípadným dosadením hodnôt.
- **kombinovanie a skladanie vzorcov** Program nevyrieši príklady, ktoré obsahujú všetky potrebné údaje, ale na ich vyriešenie by musel správne skombinovať dva alebo viac vzorcov. Tu by prichádzalo do úvahy sofistikovanejšie vyhľadávanie medzi vzorcami, ktoré by poskytlo také kombinácie vzorcov, ktoré by vyhovovali daným a požadovaným veličinám. Pri tomto by sa dala taktiež využiť rozmerová analýza, ktorá by poskytla aspoň čiastočnú nápovedu, ako údaje poskladať.
- **viacero úloh v jednej** Program sa riadi tým, ktoré údaje sú poskytnuté. Občas sa uvádzajú viaceré podúlohy, v ktorých treba použiť rôzne podmnožiny poskytnutých údajov. Program uvažuje len množinu všetkých údajov a nevie si podeliť úlohu na podúlohy. Čiastočne by sa tento problém dal riešiť gramatickou analýzou vety a formálnejšou reprezentáciou zadania úlohy.
- **slovné modifikátory** Slovné úlohy obsahujú modifikátory, ktoré čiastočne menia zadanie úlohy. Gramaticky nie sú číselného charakteru, ale napriek tomu predstavujú konštantu, ktorú treba vo vzorci explicitne doplniť. Príkladom môže byť doplnenie spojenia „a späť“ v úlohe na rovnomerný pohyb, ktoré znamená, že určenú

dráhu medzi dvoma bodmi treba zarátať dvakrát. Program takýmto spojeniam nerozumie. Teoreticky by bolo možné zvýšiť úspešnosť učením, čiže by sa naučil, ktorý modifikátor znamená (v bežnom prípade) doplnenie ktorej konštanty.

- **fyzikálna úvaha** Niektoré úlohy nevedú k použitiu vzorca, ale preverujú pochopenie látky. Napr. „Uved' príklady rovnomerného pohybu telesa“. Takéto úvahy program vykonať nevie. Na niektoré typy otázok (napr. definície) by sa mohol pokúsiť odpovedať predprogramovanou alebo naučenou odpoveďou, ak by obsahovala tie správne kľúčové slová; množstvo úloh je však veľmi špecifických a vyžaduje porozumenie realite napr. „Je pohyb rýchlika medzi dvoma stanicami vzhľadom na koľajnici rovnomerný pohyb? “ (parafrázované z [Chyba! Nenašiel sa žiaden zdroj odkazov.]).

## 5.2 Učenie

Na základe prostriedkov kategorizácie textu by bolo možné zatriediť nové úlohy do rôznych typov a naučiť sa k nim správny vzorec. Tento postup by bolo možné použiť aj pri rozhodovaní, ktorý z aplikovateľných vzorcov použiť, prípadne akú dodatočnú konštantu treba zvoliť.

## 5.3 Existujúce prístupy

Stránka [5] dokáže riešiť niektoré matematické a fyzikálne problémy formulované v angličtine, avšak tie musia byť v určenom formáte a s určenými kľúčovými slovami.

Dokáže vyriešiť napr. úlohu „What is the time of a fall with height = 125 m? “ ale nedokáže vyriešiť úlohu „How long does a stone take to fall if the stone is dropped from a cliff 125 meters above its base? “

## 6 Záver

Vytvorili sme program, schopný riešiť jednoduché fyzikálne slovné úlohy. Napriek jednoduchému princípu prekvapivo dokáže riešiť časť úloh, pričom poskytuje priestor na voľnú formuláciu úlohy, bez obmedzovania sa na poradie údajov alebo štruktúru vety.

Časť úloh nedokáže vyriešiť. Navrhujeme niektoré možné úpravy do budúcnosti, ktoré by úspešnosť zvýšili (hoci, samozrejme, stále len v priblížení, nie na 100%).

Program je zaujímavý aj z didaktického hľadiska: poskytuje vhľad do toho, ktoré úlohy majú potenciál otestovať porozumenie, a ktoré úlohy možno riešiť viacerou mechanicky.

## 7 Pod'akovanie

Tento článok bol podporený grantom VEGA 1/0439/11.

## Literatúra

- [1] Jaroslav Vachek et al. *Fyzika pre 1. ročník gymnázií*. SPN, 1984.
- [2] Jiří Bohuněk, Růžena Kolářová, Karol Klobušický, Eva Procházková. *Fyzika pre 7. ročník základnej školy: Pracovná časť B*. SPN, 1991.
- [3] Miroslav Kocun. Fyzikálna Eliza – Java Applet. <http://www.st.fmph.uniba.sk/~kocun1/BCAPV2/bin/>.
- [4] Miroslav Kocun, Michal Malý (školiťel). Fyzikálna Eliza, Bakalárska práca. Fakulta matematiky, fyziky a informatiky, Univerzita Komenského, Bratislava. Dostupné na: <https://stella.uniba.sk/zkp-storage/dzb/dostupne/FM/2010/2010-FM-nIUOmt%/2010-FM-nIUOmt.pdf>, 2010.
- [5] Novak, Gordon. Math / Physics Problem Solver. <http://userweb.cs.utexas.edu/users/novak/cgi/physdemo.cgi>, 2008.
- [6] Joseph Weizenbaum. Eliza—a computer program for the study of natural language communication between man and machine. *Commun. ACM*, 9(1):36–45, 1966.
- [7] R.P. Feynman: Cargo Cult Science. *Engineering and Science*, Volume 37:7, June 1974. Prístupné na: <http://calteches.library.caltech.edu/51/2/CargoCult.pdf>