

Др Гордана Павловић-Лажетић

РАЧУНАРСТВО У НАУЦИ И ОБРАЗОВАЊУ НА ПОЧЕТКУ 21. ВЕКА

1. Увод

Почетак новог миленијума представља изузетно динамичан период у развоју рачунарске науке и технологије. Дигитални свет постаје подршка свакодневном животу, али и основна полуга у глобализацији света. Рачунари се, с једне стране, интегришу у модерну културу, а с друге стране препознају се као водећа сила у расту светске економије. Нове технологије уводе се непрестано, и при томе застаревају готово чим се појаве [1].

Овако буран развој рачунарства омогућује пробоје у научним истраживањима других наука, нпр. генетике и биоинформатике, истраживања природног језика, итд. Тако је пројекат секвенционирања и обележавања људског генома један од најзбудљивијих истраживачких подухвата свих времена, незамислив без снажне подршке рачунарске технологије [2]. Ништа мање занимљив развој лексичко-семантичких језичких ресурса и алата, препознавање и разумевање текста и говора, концептуалног индексирања и претраживања непрегледних колекција докумената разних врста, омогућен је захваљујући савременој рачунарској технологији [3].

Поред примена у другим наукама, у живи рачунарства поново је и сама рачунарска наука. Тако је један од математичких проблема за 21. век, тзв. „проблема од милион долара“, постављених на миленијумском састанку у Паризу, 2000. године, управо везан за рачунарску науку [4]. То је проблем једнакости (или дисјунктности) класа **P** и **NP** проблема (проблем је из класе **P** ако постоји алгоритам, тј. рачунарски програм, који га решава у полиномијалном времену у односу на димензију проблема, а из класе **NP** ако је понуђено решење могуће проверити у полиномијалном времену).

Са друге стране, брзи развој рачунарске дисциплине има суштинске ефекте на образовање, и то како на образовне садржаје тако и на образовне методе. Тако су, на пример, умрежавање и веб постали критична основа рачунарске науке, а истовремено и једно од основних педагошких средстава, условаљавајући промене и у самом образовном процесу, не само у рачунарској науци већи и у другим областима. Напредак који је постигнут у технологији мултимедија на

Пленарно излагање на Републичком семинару о настави математике и рачунарства, одржаном у Београду, 12.02.2004. године

нивоу персоналних рачунара, као и у технологији умрежавања и посебно интернета и веба, створио је нове могућности за трансформисање наставног процеса и образовног система значајних размера, пре свега у развијеном свету, али и код нас. Улога наставника се у том процесу мења али и даље остаје пресудна: мада информатичка глобализација омогућује ученицима да посете далеке музеје, да праве електронске излете до далеких археолошких налазишта и да комуницирају преко видео конференција, школа је та која ученике треба да учи критичком мишљењу, социјалном понашању, радној дисциплини, личној одговорности, која их упознаје са моделима и менторима, развија и подржава радозналост.

У овом тексту биће пре свега речи о рачунарској науци и рачунарству у науци, тј. о проблемима који представљају научни изазов у самој области рачунарства, као и у другим наукама под утицајем могућности које пружа рачунарска технологија. Затим ће бити представљени неки аспекти коришћења савремених рачунарских технологија у наставном процесу, као и неки митови који се противе новој педагогији самосталног учења. Биће изложене сугестије и модели наставног садржаја у области рачунарства, на целој образовној вертикали, дефинисани од стране компетентних међународних институција [1], [5], [6] и искуства других у извођењу реформи наставног процеса, посебно у области рачунарства.

2. Рачунарска наука

Према класификацији водеће светске рачунарске асоцијације АСМ (Association for Computing Machinery) неке од кључних области које чине рачунарску науку јесу:

- Теорија рачунарства
- Подаци
- Софтвер
- Математичка израчунавања
- Методологија израчунавања
- Организација рачунарских система
- ...

Област теорије рачунарства, са своје стране, укључује, између осталих, следеће дисциплине:

- Теорија аутомата
- Математичка логика и формални језици
- Верификација
- Сложеност проблема
- ...

Задржимо се на последње две дисциплине.

2.1. Верификација

Верификација је процес утврђивања коректности алгорита, тј. програма, на основу његовог текста, без потребе за тестирањем. Дакле, верификација је *доказивање* коректности алгорита тј. програма. Основна предност верификације

над тестирањем у утврђивању коректности алгоритма јесте да је верификацијом могуће доказати коректност (али и установити евентуалне грешке), док се тестирањем можемо само у већој или мањој мери *уверити* да је алгоритам коректан. Све је већи број проблема који захтевају доказано коректна рачунарска решења (праћење летова, космички програми, итд).

Један од првих случајева који су указали на неопходност примене верификације у решавању критичних проблема (у којима се грешке у програмима мере и људским животима) описан је у [RaviSethi]. Јула 1962. године лансирана је ракета која је носила Венерину сонду без посаде, Маринер I, али је морала да буде оборена само 290 секунди после лансирања. Губитак је процењен на 18--20 милиона долара, а узрок је била грешка у програмском сегменту који је требало да обезбеди да се аутоматска корекција путање врши само док постоји контакт ракете са радаром. Али, уместо исказа

if **not** (ракета у контакту са радаром) then

не кориговати путању лета

у програму је омашком изостављена друга негација па је главни рачунар наставио са корекцијом путање и када је ракета изгубила контакт са радаром. Мада је програм претходно прошао три стотине независних тестирања, и био успешно коришћен у четири лансирања на Месец, грешка је остала неоткривена, што је довело до закључка да је неопходно предузети мере које ће обезбедити да се слична грешка никада више не понови. Овај закључак преточен је у примене метода доказивања коректности програма на основу његове интерне структуре, без потребе за тестирањем.

Доказивање коректности заснива се на два основна принципа: установљавању *програмских инваријанти*, тј. константних релација међу програмским променљивим у одређеним тачкама програма, и обезбеђивању да ће се програм завршити у *коначном* времену. Покажимо принцип доказивања коректности на примеру програма за целобројно дељење. Задатак је, за дате природне бројеве x , y , одредити целобројни део количника и остатак при целобројном дељењу x са y , тј. бројеве q , r , такве да је $x = q * y + r$, $r < y$. При томе, дозвољено је користити само операције сабирања и одузимања, па се дељење реализује узастопним одузимањем. Дакле, програм треба да произведе трансформацију $\{x, y \text{ из } \mathbf{N}\} \rightarrow \{q, r \text{ из } \mathbf{N} : x = q * y + r, r < y\}$ која се назива *формалном спецификацијом* проблема целобројног дељења. Програм се састоји од почетне доделе вредности количнику ($q = 0$) и остатку ($r = x$), за којом следи циклус у коме се понавља увећање количника за 1 и умањење остатка за вредност делиоца y , све док је то умањење могуће, тј. док остатак не постане мањи од делиоца. Може се доказати да после сваког пара радњи у циклусу важи инваријанта $x = q * y + r$, што при изласку из петље, због негације услова петље ($r \geq y$), производи тражени резултат: $q, r : x = q * y + r, r < y$. Део програма (у програмском језику C), обележен релацијама-инваријантама има следећи изглед:

```
/*   x,y in N} → {q,r in N: x=q*y+r, r<y*/
{
```

```

q=0; r=x;
while (r>=y)
/*   x=q*y+r, r>=y */
  {   r=r-y; q=q+1;   }
}
/*   x=q*y+r, r<y */

```

2.2. Сложеност проблема

За решавање једног проблема (нпр. израчунавање вредности полинома) могуће је конструисати већи број алгоритама различите временске (и просторне) сложености. Под *временском сложеностију алгорита* подразумева се функција која апроксимира број корака (операција) који ће се извршити при једном извршењу тог алгорита. Пошто тај број корака зависи од тзв. димензије проблема (нпр. од степена полинома, n), и временска сложеност алгорита представља функцију димензије проблема n . Тако за израчунавање вредности полинома познајемо алгоритме са квадратном и линеарном временском сложеностију, тј. са временском сложеностију $O(n^2)$, односно $O(n)$. Ти алгоритми рачунају вредност полинома према следећим формулама:

АЛГОРИТАМ 1. $P_n(x) = P_{n-1}(x) + a_n * x^n$

Сложеност алгорита је $O(n^2)$, јер укључује по i множења и једно сабирање, за свако i од 1 до n .

АЛГОРИТАМ 2 (ХОРНЕРОВА ШЕМА). $P_n(x) = ((\dots((a_n * x + a_{n-1}) * x + a_{n-2}) \dots) * x + a_1) * x + a_0$

Сложеност алгорита је $O(n)$, јер укључује само по једно множење и једно сабирање, у сваком од n корака.

Постоје проблеми за које познати алгоритми имају експоненцијалну сложеност која неупоредиво брже расте с порастом димензије проблема од било које полиномијалне сложености. Такав је проблем бојења графа у три боје. Проблем се састоји у бојењу сваког чвора задатог графа једном од задате три боје, тако да ниједна два суседна чвора нису обојена истом бојом. У следећем рекурзивном алгоритму за решавање овог проблема, $\Gamma(\mathbf{A}, \mathbf{R})$ је ознака за граф са чворовима \mathbf{A} и потезима (гранана) \mathbf{R} , U је скуп обојених чворова, а ознаке боја су **1**, **2**, **3**. На почетку рада алгорита претпоставља се да ниједан чвор није обојен, тј. да је скуп обојених чворова празан.

Војенје-3($\Gamma(\mathbf{A}, \mathbf{R}), U$);

```

{   if (A == U) {"bojenje završeno"; kraj;}
    else {
        izabрати чвор a који није u U;
        for (b=1;b<=3;b++)
            if (ниједан сусед од a није обојен бојом b)
                {   додати a скупу U са бојом b;

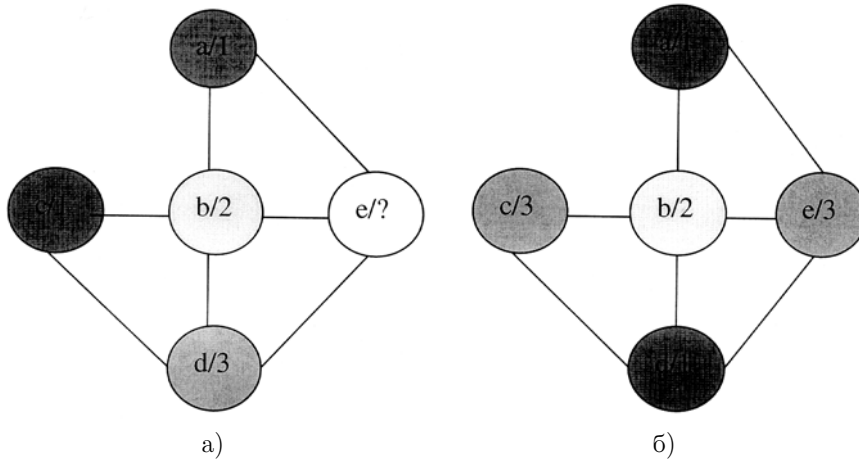
```

```

    Војенје_3( $\Gamma, U$ );
  }
}
}

```

Овај алгоритам је пример тзв. обиласка са враћањем (енгл. backtracking). Он обилази чворове у неком редоследу, и ако је чвор небојен, покушава да га обоји неком од допуштених боја (у зависности од обојености његових суседа). Ако успе, додаје чвор са придруженом бојом скупу обојених чворова и позива (рекурзивно) алгоритам за бојење преосталих чворова. Ако не успе, враћа се на претходни успешно обојени чвор (претходни ниво рекурзије) и покушава да га (успешно) обоји неком другом бојом. На пример, за граф са чворовима **a**, **b**, **c**, **d**, **e** (слика 1), који се обилазе тим редоследом, и бојама црвена (**1**), жута (**2**), плава (**3**), алгоритам ће довести прво до немогућности успешног бојења чвора **e** (слика 1а). Затим ће се вратити на претходно успешно обојени чвор **d**. Али, како је он обојен бојом **3** (што значи да није могао да буде обојен бојама **1** и **2**), алгоритам наставља са враћањем и претходно обојени чвор **c** (обојен бојом **1**) боји следећом могућом бојом **3**. Сада се, настављајући унапред, чвор **d** успешно боји бојом **1**, а чвор **e** се успешно боји бојом **3** (слика 1б).



Сл. 1. Бојење графа у 3 боје

За граф са **n** чворова претходни алгоритам има сложеност $O(3^n)$, јер у најгорем случају подразумева проверу свих 3^n могућности у распореду 3 боје на **n** чворова.

Колика је разлика у ефикасности полиномијалних и експоненцијалних алгоритама показује табела 1. Врсте означавају сложеност алгоритма (**n**, **n²**, **n³**, **n⁵**, **2ⁿ**, **3ⁿ**) а колоне димензију проблема (10, 20, 30, 40, 50, 60). У пресеку врсте **i** и колоне **j** назначено је време **t(i, j)** потребно да се алгоритмом сложености из **i**-те врсте реши проблем димензије из **j**-те колоне, ако је за решавање проблема димензије 10 (из **1.** колоне) потребно време **t(i, 1)**.

Ф-ја врем. сложености	10	20	30	40	50	60
n	0.00001 s	0.00002 s	0.00003 s	0.00004 s	0.00005 s	0.00006 s
n^2	0.0001 s	0.0004 s	0.0009 s	0.0016 s	0.0025 s	0.0036 s
n^3	0.001 s	0.008 s	0.027 s	0.064 s	0.125 s	0.216 s
n^5	0.1 s	3.2 s	24.3 s	1.7 min	5.2 min	13.0 min
2^n	0.001 s	1.0 s	17.9 min	12.7 дан	35.7 год	366 век
3^n	0.059 s	58 min	6.5 год	3855 век	$2 \cdot 10^8$ век	$1.3 \cdot 10^{13}$ век

Табела 1. Време извршавања алгоритма у зависности од његове сложености и димензије проблема

Из табеле 1 види се да за алгоритме експоненцијалне сложености (2^n , 3^n), при порасту димензије проблема време потребно за њихово извршавање недопустиво расте (са 0.059 секунди за проблем димензије 10, на пример, за граф са 10 чворова у претходном алгоритму, на 10^{13} векова за проблем димензије 60, на пример, за граф са 60 чворова). То није случај при полиномијалној сложености.

Сложеност проблема мери се сложеносту најефикаснијег алгоритма који је конструисан за решавање тог проблема. Тако је израчунавање вредности полинома степена n проблем сложености $O(n)$, а бојење графа у три боје је проблем сложености $O(3^n)$. Ако је сложеност проблема димензије n апроксимирана полиномом произвољног степена од n , $P(n)$ (нпр. n^2 , n^3 , n^{1125}), каже се да проблем има *полиномијалну сложеност* и да припада **класи P**. Такав је проблем израчунавања вредности полинома. Ако је сложеност проблема експоненцијална, нпр. 2^n , 3^n а понуђено решење проблема може се проверити у полиномијалном времену, проблем припада **класи NP**. Такав је проблем бојења графа у три боје, јер се за граф димензије n (за n чворова), за један предложени распоред боја може у n корака проверити да ли представља успешно бојење (нпр. $(a, 1)$, $(b, 2)$, $(c, 3)$, $(d, 1)$, $(e, 3)$).

Године 1971. С. Кук (S. Cook) и Л. Левин (L. Levin) поставили су питање односа класа **P** и **NP** -- да ли су дисјунктне, да ли се поклапају или преклапају. Наиме, чињеница да за проблем бојења графа у три боје није још нађен полиномијални алгоритам не значи, сама по себи, да такав проблем не постоји. Ако би био пронађен, он би „пребацио“ овај проблем из класе **NP** у класу **P**. На миленијумском састанку у Паризу, 24. маја 2000. године, Клејов математички институт из Масачусетса поставио је седам наградних математичких проблема за 21. век, тзв. миленијумских проблема са наградним фондом од по милион долара. Један од тих проблема је проблем једнакости класа **P** и **NP**, тј. утврђивања да ли је **P** = **NP**. Париз је одабран у знак сећања на Давида Хилберта (D. Hilbert) који је на Конгресу математичара у Паризу 1900. године поставио 23 отворена математичка проблема за 20. век.

3. Рачунарство у другим наукама

Буран развој рачунарства, пре свега рачунарске технологије, на почетку овога века, дао је велики допринос развоју других наука као што су математика, нуклеарна физика, биофизика, биохемија, психологија, рачунарска лингвистика, истраживања природног језика, биоинформатика, итд. Размотримо нека од достигнућа у домену рачунарске обраде природних језика и биоинформатике, која су незамислива без коришћења моћних рачунарских технологија.

3.1. Рачунарска обрада природног језика

У домену обраде природног језика, развој рачунарске технологије даје изузетан допринос низу дисциплина као што су корпусна лингвистика, електронска лексикографија, развој алата за обраду језичких ресурса, аутоматско превођење, семантички веб, препознавање и разумевање текста и говора, претраживање великих колекција докумената, екстракција информација, итд. Један од савремених електронских језичких ресурса који се развија за велики број језика, укључујући и српски (у оквиру групе за обраду природних језика на Математичком факултету у Београду), јесте лексичко-семантичка мрежа позната као **WordNet**.

WordNet је специјално структурирани речник који, уместо речи, укључује лексичке концепте, тј. значења представљена скуповима синонима (енгл. *synsets*). Над лексичким концептима успостављају се лексичке и семантичке релације (хомонимија, целина/део, надређеност/подређеност, итд), које одражавају психолингвистичке теорије лексичке меморије.

Основни циљ изградње оваквог лексичког референтног система јесте претраживање текстуалних база по семантичким критеријумима. На пример, нека је потребно издвојити делове неког текста (нпр. реченице или параграфе) у којима се помиње неко небеско тело. Стандардни речник Матице српске (МС) садржи само једну релевантну одредницу:

- **тело** с (лок. телу; мн. тела, ген. тела, и телеса, ген. телеса) **1.** биол **а.** човечји или животињски организам . . . **б.** мртав чивечји организам, леш . . . **2.а.** физ. сваки посебан предмет у простору, који може бити крут . . . **б.** засебан део неког предмета; **3.** фиг. удружење, скуп, организација, служба, установа с одређеном наменом . . . **4.** мат. (геометријско . . .) **5.** астр. небеско (свемирско) тело . . .

Ова одредница нас не упућује на термине које би требало употребити при тражењу жељене информације: ако се присетимо да су небеска тела звезде, Сунце, Марс, Земља, Месец, можемо потражити одговарајуће одреднице у стандардном речнику:

- **земља** ж (ак. земљу, вок. земљо; мн. земље, ген. земаља) **1.а.** (Земља) планета на којој живимо; **б.** место, простор живота и људске делатности, свет; . . . **2.а.** површина Земље непокривена водом, копно . . . **3.** држава или већа географска или административна област . . .

У **WordNet**-у речник је организован по концептима, тј. значењима, па ћемо и за литерал „тело“ и за литерал „земља“ имати по онолики „одредница“ колико

има значења, организованих у синсетове (скупове синонима) са другим литералима у синонимним значењима:

- ⟨тело 1, организам 1 (човечји или животињски организам, ...)⟩
- ⟨тело 2а, предмет 1, ... ⟩
- ⟨тело 3, удружење, скуп 1.а, организација 2, служба 3, ... ⟩
- ⟨тело 4, мат (геометријско)⟩
- ⟨тело 5, астр. небеско (свемирско)⟩

Сваки од ових концепата је мрежом релација повезан са другим концептима, па концепт који се односи на небеско тело заправо има под собом хијерархију тзв. хипонима (подређених концепата) а над собом хијерархију тзв. хиперонима (надређених концепата):

- ◁ ⟨ентитет, објекат 1 ... ⟩
 - ◁ ⟨предмет, неживи објекат ... ⟩
 - ◁ ⟨природни објекат, ... ⟩
 - ◁ **⟨тело 5, астр. небеско (свемирско)⟩**
 - ◁ ⟨планета 1, ... ⟩
 - ◁ ⟨Земља 1. -- (планета на којој живимо)⟩
 - ◁ ⟨Меркур 1 ... ⟩
 - ◁ ⟨Венера 1 ... ⟩
 - ◁ ...
 - ◁ ⟨звезда 1, ... ⟩
 - ◁ ⟨Сунце 1 ... ⟩
 - ◁ ...
 - ◁ ⟨сателит 1, ... ⟩
 - ◁ ⟨Месец 1, ... ⟩
 - ◁ ...

С друге стране, и сваки концепт који укључује различита значења литерала „земља“ имаће припадну мрежу релација, на пример:

- ◁ ⟨ентитет, објекат 1 ... ⟩
 - ◁ ⟨предмет, неживи објекат ... ⟩
 - ◁ ⟨природни објекат, ... ⟩
 - ◁ ⟨тело 5, астр. небеско (свемирско)⟩
 - ◁ ⟨планета 1, ... ⟩
 - ◁ **⟨Земља 1. – (планета на којој живимо)⟩**
 - ◁ ⟨место 1а, ... ⟩
 - ◁ ⟨предео 1 ... ⟩
 - ◁ ⟨срез 1, округ 1, област 2 ... ⟩
 - ◁ ⟨административно подручје 1⟩
 - ◁ **⟨земља 3 (држава или већа географска или административна област) ... ⟩**

Организација **WordNet**-а омогућује читав низ примена које се не могу ни замислити без оваквог језичког ресурса.

3.2. Биоинформатика

Биоинформатика је једна из скупа фундаменталних или примењених биолошких научних дисциплина које свој бурни развој у овом веку дугују, између осталог, бурном развоју рачунарске технологије. Између осталих, ту су и биотехнологија, генетика, функционална геномика, биосинтеза, протеиномика, медицина, фармација, итд.

Основни објекат изучавања биоинформатике јесте геномска секвенца (ланац ДНК), тј. „текст“ над четворословном азбуком нуклеотида -- пуринско-пиримидинских база, **A** (аденин), **C** (цитозин), **G** (гуанин), **T** односно **U** (тимин односно урацил). Свака тројка нуклеотида (тзв. кодон) кодира једну од 20 аминокиселина према шеми приказаној на слици 2, а аминокиселине низањем граде протеине.

		Second base of codon				
		U	C	A	G	
First base of codon	U	UUU } Phe UUC } UUA } Leu UUG }	UCU } UCC } SER UCA } UCG }	UAU } Tyr UAC } UAA } UAG }	UGU } Cys UGC } UGA } UGG } Trp	U C A G
	C	CUU } Leu CUC } CUA } CUG }	CCU } CCC } Pro CCA } CCG }	CAU } His CAC } CAA } Gln CAG }	CGU } Arg CGC } CGA } CGG }	U C A G
	A	AUU } Ile AUC } AUA } AUG } Met	ACU } ACC } Thy ACA } ACG }	AAU } Asn AAC } AAA } Lys AAG }	AGU } Ser AGC } AGA } Arg AGG }	U C A G
	G	GUU } Val GUC } GUA } GUG }	GCU } GCC } Ala GCA } GCG }	GAU } Asp GAC } GAA } Glu GAG }	GGU } GGC } Gly GGA } GGG }	U C A G

Сл. 2. Кодирање аминокиселина (Phe, Leu, ...) кодонима нуклеотида

Основни задаци биоинформатике су секвенционирање (генома разних организама), рачунарске анализе сегмената генома, идентификација гена (секвенци које кодирају протеине), класификовање, кластеровање, израда генских мапа, одређивање структуре протеина, мутација, патогенеза, итд.

Основна средства биоинформатике су статистичке анализе расподеле слова, биграма (парова), триграма (тројки), заступљености кодона, ентропије, анализа зависности, итд.

Један од најзбудљивијих научних пројеката свих времена који је посебно допринео савременом развоју биоинформатике јесте пројекат секвенционирања хуманог (људског) генома са трајањем од 1990. до 2005. године. У овом пројекту

учествује преко 50 лабораторија САД, Европе и Јапана а циљ је да се секвенционира -- издвоји и обележи сваки од 24 људска хромозома и да се установи позиција гена. Године 2001. публикован је први нацрт људског генома у часописима Nature и Science. Добијени резултати указују на постојање 25000--40000 гена у људском геному, као и на чињеницу да само 3% укупне генске секвенце (ланца ДНК) представљају гени, тј. секвенце које кодирају протеине. Ови резултати представљају функционалну скицу и еволуциону историју људске врсте.

Непосредни значај биоинформатике за човека, природу и квалитет живота је велики и зато се у ову област улажу огромна средства. Резултати биоинформатике доприносе побољшаној дијагностици болести, откривању генетске предиспозиције болести, изградњи специфичних лекова на основу индивидуалних генетских профила, употреби генске терапије, раном откривању и третману патогенеזה, идентификовању ДНК, итд.

4. Образовање

Нови миленијум доноси промене, пре свега технолошке, које се нужно одражавају на наставни процес. Те промене су двојаче -- како еволутивне, које трају већ много година, тако и револуционарне. Као илустрација еволутивних промена, може се поменути предвиђање оснивача Intel корпорације Гордона Мура (G. Moore) из 1965. године да ће се густина микропроцесорског чипа удвостручавати сваких осамнаест месеци, које се остварује и данас. Зато експоненцијални раст расположиве рачунарске снаге омогућује решавање проблема који су били недоступни до пре само неколико година. Као илустрацију револуционарних промена поменимо брзи раст умрежавања после појаве веба (World Wide Web). Мрежне технологије постају и наставни садржај и суштинско педагошко средство. Драматични пораст коришћења рачунара широм света повећава општу информатичку информисаност, али и јаз између ученика који користе и оних који не користе рачунаре. Долази до прихватања рачунарске науке као академске дисциплине и њеног ширења.

Посебан значај добијају курикуларне теме као што су:

- веб и примене;
- мрежне технологије, посебно оне базиране на **TCP/IP** протоколу;
- графика и мултимедија;
- релационе базе података;
- објектно оријентисано програмирање;
- коришћење **API**;
- безбедност софтвера;
- заштита и криптографија.

Као циљеви у настави саме рачунарске науке постављају се:

- увођење свих ученика у принципе и методологију рачунарске науке;
- заснивање рачунарске науке као академске и професионалне дисциплине, укључујући и разлике између рачунарске науке и информационе технологије, информационих система, математике, других наука;

- овладавање научном и математичком димензијом рачунарске науке;
- овладавање практичном димензијом рачунарске науке (рад са текстом, звуком, видеом; различите могућности у пројектовању алата).

Према резултатима анализе садржаја и значаја појединих рачунарских области и тема, коју детаљно спроводи АСМ, предлог програма предмета који се односи на рачунарску науку у **средњој школи** укључује седам области:

1. Алгоритми
2. Програмски језици
3. Оперативни системи и корисничка подршка
4. Архитектура рачунара
5. Друштвени, етички и професионални контекст
6. Примене рачунара
7. Додатне теме

При томе, свака од области има основне, препоручене и опционе теме. У области алгоритама, на пример, основне теме су:

- 1.1. Алгоритми у реалности
- 1.2. Технике пројектовања и представљања алгоритама
- 1.3. Примери важних алгоритама
- 1.4. Основни концепти решавања проблема, нпр. итерација, рекурзија

У препоручене теме спадају:

- 1.5. Методе тестирања алгоритама
- 1.6. Основне структуре података
- 1.7. Булова алгебра

док опционе теме укључују:

- 1.8. Коректност, коначност алгоритама
- 1.9. Сложеност алгоритама
- 1.10. Границе израчунљивости

Слично, у област програмских језика укључују се основне теме:

- 2.1. Увод у специфични програмски језик
- 2.2. Концепт секвенце, селекције и репетиције
- 2.3. Нивои програмских језика

Препоручене теме укључују:

- 2.4. Компилаторе и интерпретере
- 2.5. Модуларизацију програма -- функције и процедуре
- 2.6. Анализу готовог програмског кода

Опционе теме програмских језика су:

- 2.7. Поређење језика: процедурални, структурни, функционални, објектно-оријентисани, паралелни
- 2.8. Верификација програма, пред и пост услови, инваријанта петље
- 2.9. Теоријске машине и формални језици

Информатичко образовање у **основној школи**, према АСМ-овим препорукама, укључује:

- информатичке технологије (кроз разне курсеве);
- припрему за курс рачунарске науке у средњој школи: елементе логичког и алгоритамског мишљења;
- многе информатичке активности које се изводе без рачунара, на пример:
 - бинарно представљање цртежа (црно-белим квадратићима);
 - сортирање (речи исписаних на папиру);
 - претраживање означених картица (линеарно, бинарно, хеш);
 - игра примања и предаје лоптица суседима -- симулација мрежног саобраћаја, итд.

5. Уместо закључка

Уместо закључка, наведимо неке карактеристике промена образовног процеса до којих неминовно долази под утицајем рачунарске технологије на почетку новог века и новог миленијума.

- Нова педагогија самосталног учења последица је све већег коришћења интернет технологије и рачунара уопште, као и дистрибуираних могућности приступа дидактичком материјалу, и његове размене. Интернет и веб се јављају као дидактичка средства, као што је то у прошлом веку био радио и филм.
- Неки митови који се противе новој педагогији самосталног учења осуђени су на рушење. Такав је мит о постојању јединственог тачног одговора, као и мит о томе да на свако питање постоји, негде, експлицитно записан одговор. Интернет би данима могао да се претражује безуспешно, на пример, у трагању за експлицитним одговором на питање која је вулканска купа на највећој надморској висини, али се до одговора вероватно може стићи само истраживањем, анализирањем и поређењем података о разним вулканским купама.
- Информатичка глобализација омогућује мултимедијални приступ удаљеним локалитетима као што су музеји и археолошка налазишта, и учествовање у новим електронским видовима дружења и комуницирања.
- Нове могућности за трансформисање образовног система ширих размера створене су технологијом мултимедија на нивоу персоналних рачунара, као и технологијом умрежавања и посебно интернета и веба.
- Улога наставника у процесу трансформације школе мења се али и даље остаје пресудна: они треба да уче критичком мишљењу, социјалном понашању, радној дисциплини, личној одговорности, да развијају и подржавају радозналост.

Литература

[1] Computing Curricula 2001, Chapter 3: Changes in the Computer Science Discipline, <http://www.acm.org/education/>

-
- [2] Human Genome P, The National Humane Genome Research Institute, <http://www.nhgri.nih.gov>
- [3] Text, Speech and Dialogue, Lecture Notes in AI, Springer Verlag, 2003.
- [4] A. Jackson, Million Dollar Mathematics Prizes Announced, Notices of the AMS, 9 (2000), 877–879.
- [5] S. Merritt et al, ACM Model High School Computer Science Curriculum, Task Force of the Pre-College Committee of the Education Board of the ACM.
- [6] A Model Curriculum for K-12 Computer Science: Final Report of the ACM K-12 Education Task Force Curriculum Committee, October 22, 2003, Allen Tucker (editor).