Az autodidakta/kritikus tanulás mintázatai – I. rész

(Patterns of the autodidactic/critic-oriented learning – Part I)

Pitlik László, KJE/MY-X team

Kivonat: A cikk kísérletet tesz arra, hogy világos határvonalakat húzzon azon quasi eretnek kérdés kapcsán: mi is kell, hogy a reális célja legyen a BPROF (üzemmérnök informatikus) képzésnek a BSC/MSC/PhD-szintekkel, ill. az ECDL-alapokkal összevetve? A kérdés azért eretnek, mert természetesen bárki bármiről tud, az hasznos, de a Bologna-folyamatok kezdete óta nem ez a kérdés. A kérdés az, hogy kell-e, hogy tudjon adott jelenségről egy kiképzendő ahhoz, hogy potenciális feladatait sikeresen legyen képes elvégezni diplomával a zsebében? Más megfogalmazásban: mennyit célszerű raktárra tanulni? A cikk az autodidakta tanulást és a kritikai szövegértelmezések lehetőségeit alapul véve, példák sorozatán keresztül igyekszik bemutatni egy 2009-es kanonizált és önmagában kiváló jegyzethez képest, mik azok a területek, melyekről úm. elég hallani/tudni, elég egy minden segédeszköz felhasználásával megírt (tűzvédelmi teszthez hasonló) log-állományt produkálni, s mik azok a területek, ahol érdemi feladatai vannak (az LLL keretében) a kiképzendőknek? Az LLL-re utalás azt jelenti, hogy a raktárra tanulás helyett, mely hatása életre szóló, a raktárhiány prompt feltöltésének képességét kell inkább kialakítani. Emellett a cikk arra is példát kíván felmutatni, miként lehetséges a jövőbeli (4 félévet megfejelő 2-féléves) specializációk egyikének (jelen esetben a block-chain fogalmának) a fokozatos életre keltése a kanonizált, első két évet érintő tananyagokból kiindulva. A cikk vitairat, vagyis érveken keresztül egyes kizárásra/minimalizálásra javasolt területek bekerülése nem kizárt a BPROF-kánonba és a már befogadott részek elhagyása/aránymódosítása is lehetséges valamilyen alternatívával szembeni versenyben – különös tekintettel a kiképzendők személyes igényeire. A határvonalak meghúzásakor egyrészt fontos volt jelezni, hogy mindenki, vagyis a Megrendelők is felelnek az informatikai megoldások minőségéért, hatékonyságáért, ill. azt is ki kell emelni, hogy az üzemmérnöknek mindenkivel, mindenről tárgyalnia kell tudni, de önállóan csak rel. szűk problémakört kell felelősen kezelni tudni. Tananyaggá csak az válhat, ami az azonnali éles/valós kipróbálás esélyét garantálja, vagyis ami valakinek ma a munka világában valóban reális feladatként merül fel – lehetőség szerint minél gyakrabban. A valóságtól elszakított vegytiszta példák egyike sem érték addig, amíg a valósághoz való kötődés legalább olyan részletességgel nem kerül megadásra, mint magában a vegytiszta problémakezelésnek a részletgazdagsága. A cikk hajtóerejét az a közkeletű vélekedés adja, hogy a megtanított jelenségek jelentős arányban azért vesznek el a tanulók számára, mert semmihez nem lehet kötni ezeket a diák valóságélményei kapcsán hosszasan, vagy akár soha az életben. Ez a hétköznapi kritika sajnos nem lehet teljesen alaptalan, így a kinek-mikor-mit érdemes megtanulni kérdésre optimális választ keresni nem eretnekség, hanem a tudásmenedzsment komolyan vétele – duális képzésként értelmezve a jogilag annak nem minősülő képzéseket is – az egyetem, mint tanműhely elv mentén…

Kulcsszavak: tudásmenedzsment, oktatásdidaktika, gyakorlat-orientáltság, duális képzés, tantervkészítés, akkreditáció, optimalizálás, minőségmenedzsment

Abstract: The paper tries to derive clear borders concerning a quasi heretic question: What kind of knowledge should really be transferred towards the Students concerning a BPROF degree compared to BA/BSC, MA/MSC, PhD, and ECDL levels? The question is heretic, because each knowledge element can be seen as useful, but here and now, we need a new point of view – especially after the starting of the Bologna process. The basic question can be reformulated: What kind of knowledge is needed for working really with them? Or even: Is it worth to learn about something where the knowledge elements will be stored but maybe never used? The paper will present a lot of details based on an excellent script from 2009 – details, being informative enough to declare what kind of knowledge elements should only be understood in a prompt way (see testing fire management competences in enterprises based on test where all background sources can be used for selecting the best option without any time limitation). On the other hand: what kind of knowledge elements are organic part of the visions about real jobs. The Life-Long-Learning-based teaching has to ensure, that the Students will be able to acquire new knowledge life long, quasi alone, and in a prompt way – compared to the teaching solutions, where knowledge elements (being probably needed - later in the life cycles) will be transferred at once. Parallel, the paper will try to demonstrate, how it is possible to prepare a specialization (it means the last 2 semesters in frame of a 6 semester curricula). This planned specification is the blockchain, which keyword should be identified in each possible knowledge elements before. The paper can be seen as a kind of discussion paper where the here presented preferences and aversions (incl. their ratios) can be changed based on rational argumentations and first of all: based on the specific needs of the affected individuals. The borders, the sets of the knowledge elements, their chronology and/or ratios try to declare that the customers/stakeholders of the projects concerning informatics have a massive responsibility to acquire IT-knowledge on a level ensuring an effective and efficient communication – instead of (just) dreaming about solutions. A BPROF-Student should be capable of interpreting quasi each customer-oriented approach but this level of experts in an alone-modus should have a limited responsibility concerning complex IT-systems. A knowledge element can only become a real learning material if the job-dependencies are trivial given. Extracted, pure theoretical tasks are irrelevant, or they should be finetuned with details about the using specifications. The paper believes in a deep way that the common interpretations of the society about superfluous (never used) knowledge elements should be right. So, the quasi heretic approaches of this paper is rather a reformulation of a lot of knowledge management/quality management issues, where each curricula will be evaluated as a kind of dual education – independent from the official rules about it.

Keywords: knowledge management, didactics, praxis-orientation, dual education, curricula, accreditation, quality management, optimization

# Bevezetés

A cikk apropóját a Házy Attila és Nagy Ferenc (2009): Adatstruktúrák és algoritmusok, <https://www.tankonyvtar.hu/hu/tartalom/tamop425/0046_adatstrukturak_es_algoritmusok/adatok.html> tankönyv és a BPROF képzés akkreditációs anyaga adja.

A BPROF-képzés (üzemmérnök informatikus képzés) BSC-szintű informatika képzés. Minden egyes tananyagegység esetén világos képet kell adni a Hallgató számára arról, mi az, ami quasi középiskolai szintű tudás (vö. ECDL-analóg), mi az, ami felsőfokú, de alapozó (BSC/BPROF), ill. mi az, ami már MSC szintet jelent. Sőt, azt sem árt világossá tenni, mit jelent az MSC-hez képest a PhD-szint?!

Kiindulásként vegyünk máris egy példát. Ha az akkreditált kulcsszókészlet tartalmaz olyan szakkifejezéseket, mint pl. Isten-objektum (pl. a programozási alapelvek és módszerek tantárgyhoz kötődően), akkor középiskolás szintnek tekintjük, ha valaki legalább már készített egy makrót. BSC/BPROF-szintnek tekintjük, ha képes adott új kulcsszó (Isten-objektum) kapcsán a kereséssel feltárható szakirodalmak értelmezésére, s ennek tanúbizonyságaként racionális tesztkérdések kialakítására, ahol az is fontos, miért is racionálisak didaktikai szempontból az ún. rossz válaszok?! MSC-szint lehet a KNUTH-i elv teljesülése felé való elmozdulás, amikor is valaki képes szakértői beszélgetések/szakirodalmak alapján pl. olyan szakértői rendszert alkotni (manuális szabályképzésre támaszkodva), mely alapján egy programról eldönthető feketén/fehéren, hogy Isten-objektum-e, ill. levezethető egy indexérték arra vonatkozóan, mennyire Isten-objektum egy adott objektum (kód/megoldás/program/folyamatábra)?! A PhD-szint pedig egy olyan algoritmus kialakítását tételezi fel, mely algoritmus képes előzetes pl. induktív szabálygenerálás alapján nyers forráskódokról megmondani, a fenti két kérdésre a választ (vö. igen/nem, ill. mennyire). Részletek/finomhangolások:

* <https://hu.wikipedia.org/wiki/Isten_oszt%C3%A1ly>
* <https://hu.wikipedia.org/wiki/V%C3%A9rszeg%C3%A9ny_t%C3%A1rgyk%C3%B6ri_modell>
* <https://hu.wikipedia.org/wiki/Objektumorient%C3%A1lt_programoz%C3%A1s>
* <https://en.wikipedia.org/wiki/Procedural_programming>

(A knuth-i elv alapján megfogalmazott feladat (BSC/BROF-szint = a fogalmak használni tudásának szimulálása tesztben, beszédben /// MSC = az alábbi feladat): A fenti WIKI-szócikkek alapján alkosson manuálisan előtervezett szakértői rendszert az objektum-orientált és a procedurális programozás megkülönböztetésére adott kódra vonatkozóan! Alkosson szakértői rendszert az Isten-osztály/objektum, ill. a vérszegény tárgyköri modell fogalmainak értelmezésére, vagyis adott kód esetén mikor mondható arra rá, hogy ilyen vagy olyan, ill. nem ilyen, vagy nem olyan? PhD-szint: alkosson olyan szakértői rendszert (pl. induktív módon, azaz kódokból, mint tanulási mintákból), mely automatikusan egy tetszőleges kódra vonatkozóan tudja használni/elvetni a fenti fogalmakat!)

Az adatszerkezetek és algoritmusok tantárgy szintén a BPROF-képzés akkreditációs elvárásainak megfelelően része az üzemmérnök informatikussá válásnak.

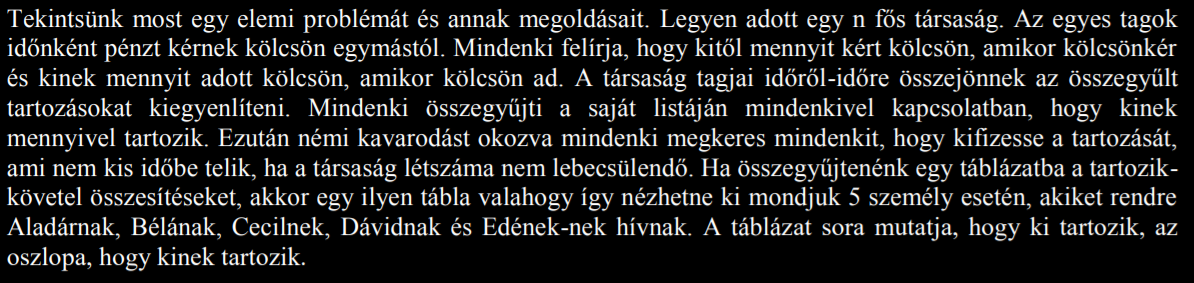
Az üzemmérnök informatikusnak képesnek kell lennie sosem látott/hallott/tapasztalt problémák esetén is autodidakta módon alkalmazkodni a kihívásokhoz, ill. ismét csak képesnek kell lennie az önfejlesztő tanulás keretében feldolgozott tetszőleges források kritikai értelmezésére is. Így jutva el legalább egy lehetséges megoldásig adott probléma esetén.

Az MSC-szint didaktikailag a BSC/BROF-szinthez képest annyi többletet jelent „csak” látszólag, hogy a mester egy lehetséges megoldás helyett kényszerűen többet kell, hogy értelmezzen annak érdekében, hogy ezek alapján az optimálisat ki tudja választani, sőt, le tudja vezetni a már létező megoldásokhoz képest új/önálló megoldásként (vö. genetikai potenciál).

A PhD-szint olyan problémákra ad választ, melyekre még semmilyen megoldás nem létezik, de itt is érvényes – nem illik megállni az első lehetséges megoldásnál, hanem sok alternatíva közül illik a legjobbat (pl. legegyszerűbbet) feltárni tudni.

# 1. Példa – Alternatív modellek

Ha az adatszerkezetek és algoritmusok tantárgy egy véletlenszerűen online elérhető jegyzetét letölti valaki (vö. Házy Attila és Nagy Ferenc (2009): Adatstruktúrák és algoritmusok), akkor már az első példa megfelelő értelmezési keretet ad a kritikai szemlélet és az autodidakta tanulás bizonyos mintázatainak felismerésére (vö. 1. és 2. oldal)[[1]](#footnote-1):

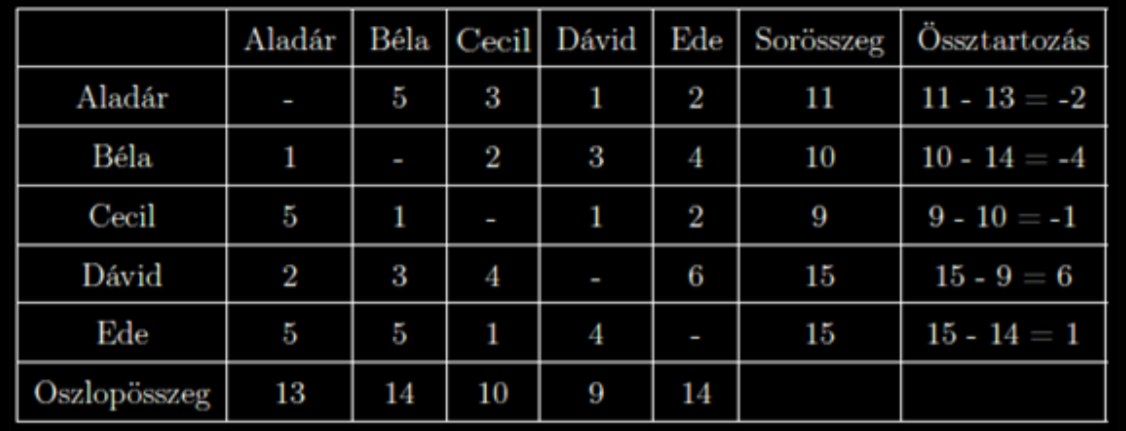


Kritikai megjegyzések:

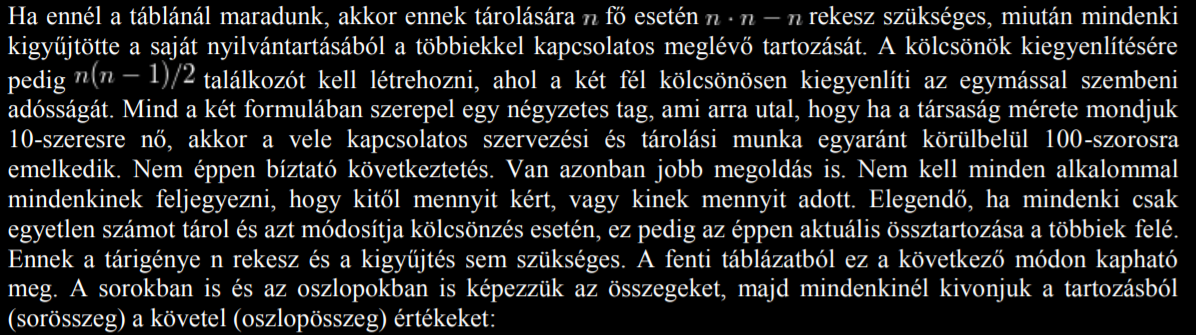
* A probléma közel sem elemi, mert a szituációból számos önálló elemi probléma/cél vezethető le pl.
  + hogyan lehet fejben végzendő műveletek nélküli elszámolási rendszert alkotni?
  + hogyan lehet az elszámoláshoz vezető találkozások számát minimalizálni?
  + hogyan lehet a téves részeredményeket kezelni? (vö. blockchain)
  + …
* A probléma elemei, vagyis a tranzakciók teljesen elhanyagolásra kerülnek, ill. a ki kinek mennyivel tartozik kérdés nem összegyűjtéssel válaszolandó meg, azaz nem darabszámos jelenség…
* Kellően PC-e, ha csak egy hölgy szerepel az 5 szereplős mintában?
* …

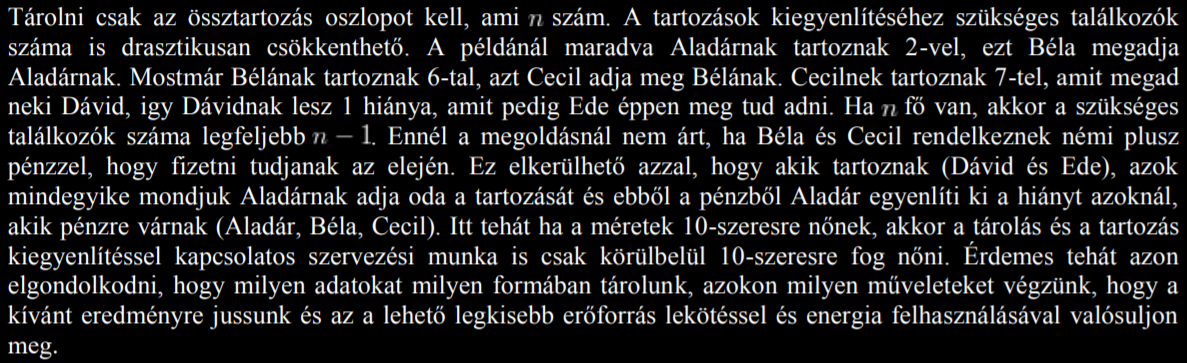
Autodidakta értelmezések:

* Vajon a tábla és a táblázat szinonimák-e?
* Mi az elemi probléma definíció? Sok-sok példa kellene az elemi és nem elemi esetekre, melyek leíró adatai alapján a klasszifikációs ellenbizonyítás sikeresen elvégezhető (vö. KNUTH: tudás az, ami forráskódba átírható, minden más pl. szómágia).
* …



A szóban forgó táblázat a ki kinek mikor mekkora összeget adott át adatok, azaz a tranzakciós adatvagyon alapján egy riport (vö. pivot, kimutatás), mely relatíve nagy tranzakciósszám esetén fejben, hibátlanul nem triviális, hogy előáll. S ha a hibák ráadásul még nem is véletlenszerűek, hanem esetleg tudatosak (vö. bizánci tábornokok probléma: <https://en.wikipedia.org/wiki/Byzantine_fault>, vö. német verzió, ill. <https://fintechradar.hu/penz/0814/mi-a-koze-a-bizanci-tabornokoknak-a-kajakiszallitashoz/>, <https://en.wikipedia.org/wiki/Dining_cryptographers_problem>), akkor annak feltárása, ki hazudik, egy új aspektust nyit meg (vö. blockchain).





Autodidakta megjegyzések:

* A rekesz fontos kulcsszónak tűnik – (keresni kell rá a teljes jegyzetben: összesen 19 említés)…
* A cél mibenléte (legkisebb erőforrás-lekötés/energia-felhasználás) fontosnak tűnik…
* A speciális keretfeltételek (pl. van-e/lehet-e több pénz a rendszerben, mint amennyi a felek között forog) alapjaiban határozza meg egy-egy megoldás értékét…
* Ha több megoldási alternatíva is van (vö. „ennél a megoldásnál”), akkor ezek minden tulajdonsága fontos és tételesen illik ismerni/feltárni tudni (vö. kritikai aspektusok – pl. <http://miau.my-x.hu/miau/185/occams_razor_finetuned.doc>)
* …

Kritikai aspektusok:

* Vélelmezhetően hamis/félrevezető állítás, hogy „nem kell minden alkalommal mindenkinek feljegyezni, hogy kitől mennyit kért, vagy kinek mennyit adott. Elegendő, ha mindenki csak egyetlen egy számot tárol…” – vagyis az összes tartozását mindenki felé. Az ’n’ növekedésével ennek a részeredménynek a következetes képzési egyre inkább a biológiai lehetetlenség irányába hat (vö. felejtés, téves számítások egymást csak ritkán kioltó sorozata).
* Az adott pillanatban való elszámoláshoz egyetlen találkozó sem kell, csak egy likvid „bank”, ahol azok, akik tartoznak befizetnek és azok, akik jogosultak adott összegre ezt megkaphatják, vagyis ’n’ db ügyfél:bank találkozóra van csak szükség, ha a találkozók minimalizálása fontos.
* Az erőforrás/energia-lekötés minimalizálása itt nem válik még mérhető valósággá, mert nem világos, a sok-sok alternatív megoldást (objektumok) milyen szempontok (attribútumok) mentén lehet összehasonlítani, s ezen attribútumértékek objektumonként pontosan ki által és hogyan tárhatók fel hibátlanul?
* Ha kell egy bank, akkor ennek létét/költségeit (egy hatékonyabb elszámolási rendszer működését) mi fedezi?
* …

Az első példa kapcsán világossá válhatott, hogy az adatvesztés (vagyis a tranzakciók rendszerszintű kezeléséről való lemondás) árán lehet egy fajta hatékonyságról beszélni. Hasonlóképpen a hatékonyság egy látszólag másik aspektusa az igazságban/őszinteségben/helyességben való bizalom léte. Ezek a speciális feltételek vélelmezhetően túlságosan nagy árat jelentenek, vagyis az univerzalizálhatóságot túl nagy veszteségek érik.

Ezzel szemben egy log-alapú (tranzakció-alapú) megközelítés riport-orientált részeredményekkel más fajta hatékonyságot mutat fel: nem kell semmilyen speciális feltétel, az összes felmerülő adat egyszerűen tárolható, s a riportálás is rutin-feladat immár. A potenciális ’n’ érintett kapcsán nem is kell, hogy mindenki mindenkit ismerjen, mert a nem létező tranzakciók értéke helyesen nullaként értelmezhető. Az ’n’ ügyfél lényegében anélkül lehet a többiek felé korrekt, hogy a teljes rendszerről bármilyen tudomása lenne, s bármilyen részeredmény helyessége/léte tőle függene, azaz bármilyen erőforrás-áldozatot kellene vállalnia – feltételezve, hogy minden tranzakció helyesen került rögzítésre.

Amennyiben felmerülhet annak gyanúja, hogy valaki nem akar őszinte lenni, akkor a probléma új dimenziót nyer. Kiindulásként tételezzük fel, hogy mindenkinek kötelessége önbevallás-jelleggel jelezni kitől mennyit kapott, kinek mennyit adott…

Ideális esetben itt most egy xls-re mutató link jönne, ahol egy demo-elszámolási rendszer szerepel blockchain-es logikákat bemutatva…

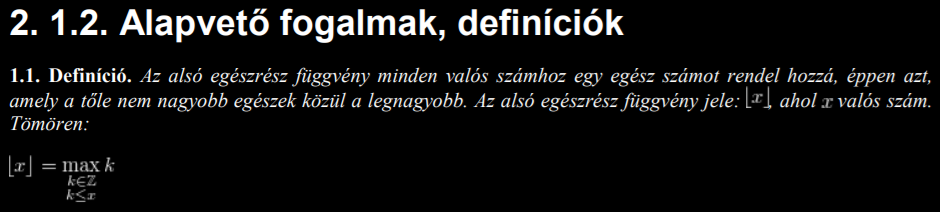
# 2. Példa – Holisztikus szemlélet (cél és elvárás)

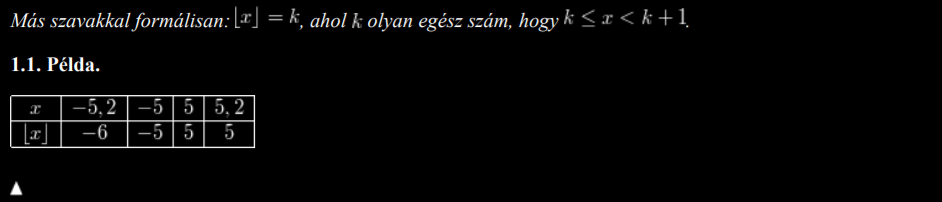
# Egy üzemmérnök informatikus nem folytathat strucc-politikát: nincs olyan, hogy ő csak egy adott szinten (vö. alábbi részlet) kíván aktív lenni:



Az üzemmérnöktől az őt alkalmazó környezet elvárja a potenciális alkalmazói együtt gondolkodást éppúgy, mint a megszületett döntések realizálását a szoftverek és hardverek, ill. ezek együttműködése tekintetében is. A BPROF-végezettség a valódi élethelyzetek üzemszerű kezelését várja el, s nem az ideális Megrendelő által kiadott ideális parancsok ideális végrehajtását, mert ez a i-i-i-konstelláció lényegében nem létezik.

# 3. Példa – A core business még sem core business





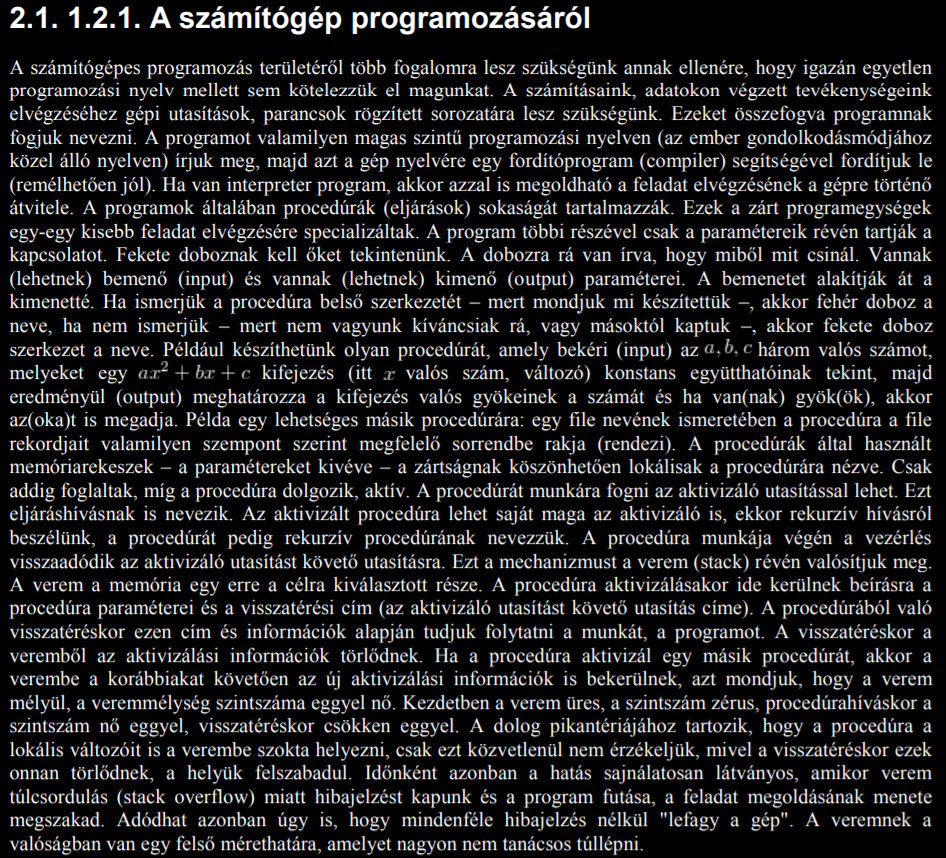
A szóban forgó jegyzet/tankönyv definíciói és ezek operatív szintre való konvertálását támogató példái (vö. KNUTH) látszólag a legértékesebb részei a tanulási/tanítási folyamatoknak. DE: ezek nem többek, mint pl. az Excel Súgója, ahol adott függvény leírása egyszerre kínál fel elméletibb és gyakorlatiasabb megközelítéseket, melyek közül sem az elméleti, sem a gyakorlati rétegek NEM tanulandók meg, „csak” úm. annyit kell a Hallgatónak önellenőrzés keretében saját magán tesztelnie, hogy azt meri-e remélni adott pillanatban, hogy megértette ezen triviális üzenetrétegeket. Ezek a core üzenetek tehát annak ellenére nem magolandók be és nem tekinthetők minimális elvárásként az oktató, az oktatási rendszer részéről, hogy minden félreértés ezen a szinte adott pillanatban egy projekt/szándék számára lehet letális. Raktárra tanulni ugyanis felesleges: a megértés öntesztje után már csak az a fontos, HOL lehet ezeket a tételes támogatásokat újra fellelni?! (Példa: egyetlen egy orvostól sem várható el, hogy fejből és hibátlanul tudja az összes gyógyszer összes létező paraméterét, de tudnia kell, hol találhatók ezek, ill. adott (először felismert és kezelni kívánt betegség esetén) miként tudja a gyógyszer-alternatívákat feltárni és egymással szemben versenyeztetni eltérő célrendszerek mentén: pl. olcsóság, robosztusság, stb.). A BPROF-képzés, de már ideális esetben a középiskolai képzés is erre a megoldási alternatívák feltárását és értelmezését/értékelését jelentő folyamatra (vö. kompetencia) kell, hogy felkészítsen mindenkit.

# 4. Példa – Mennyire kell érteni a szakmai szómágiát?

A szakszavakkal való kommunikáció minimálisan hatásos szintje két elvárásnak kell, hogy megfeleljen: egyrészt a felénk áramló szómágiát a saját magunk számára releváns részletek mélységéig érteni illik, másrészt tudni kell úgy megszólítani alacsonyabb és magasabb szintű tudással rendelkezőket, hogy ők a lehető leghatékonyabban értsék, mit is várunk tőlük.

Az autodidakta tanulás egyik sarkalatos pontja tehát a létező szakmai szövegek szókincsének feldolgozási képessége: vagyis tudni kell különbséget tenni szakszó és általános szómágikus alapanyagok között. Az alábbi példa lényegében összefoglalja a programozás jelenségét egy, már munkavégzésre is alkalmas komplexitással.

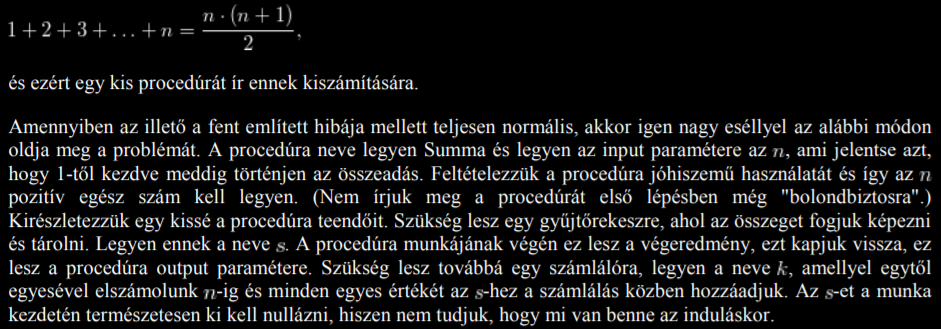
Az idézet zárógondolata értelmében: egy logikailag helyes program is vezethet sikertelen megoldáshoz. S ezzel el is jutunk a stratégiai és az operatív szemléletmódok határvonalához és nem mellesleg pl. az ugyanazon feladatot eltérő módokon megoldó program-alternatívák potenciális értékelésének jelenségéhez, mely jelenség végig fogja kísérni a teljes BPROF-képzést…



# 5. Példa – Hogyan keletkeznek alternatív megoldások?

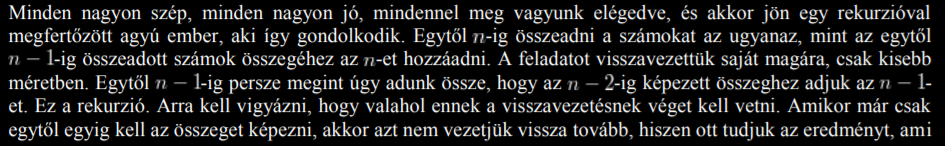
Az ideális, a laikus, s az eltérő rekurziós logikát alkalmazó megoldások halmazára a vizsgált jegyzet az alábbi példát hozza fel:





Az első megjegyzés az kell, hogy legyen, hogy NEM feltétlenül az informatikus feladata az n\*(n+1)/2 egyszerűsítés ismerete, felvetése, mert ezt MINDENKI, azaz a potenciális Megrendelők is tanulták már akár általános iskolai szinten, hiszen a mögöttes know-how egyszerű: adjunk össze ’n’ db egymás követő számot 1-től kezdődően úgy, hogy mindenkor a legnagyobbat és a legkisebbet adjuk össze (n+1)… A potenciális Megrendelők tehát ne érezzék magukat felmentve az együtt gondolkodás kényszere, a hatékonyságért való közös felelősség alól még akkor sem, ha a korábbi elvek egyike szerint az üzemmérnök informatikus sem kapszulálhatja magát semelyik szinten…

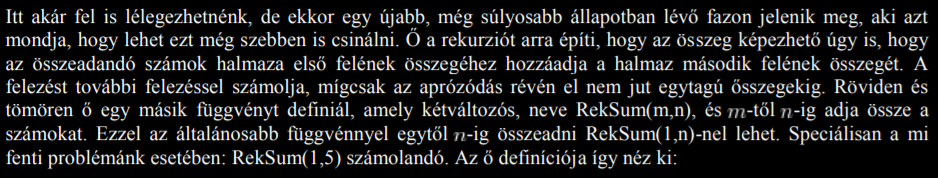
A vizsgált jegyzet felvet az előző ideális és laikus megoldás mellé még további két rekurziós megoldást, ahol a rekurzió fogalmát (az ott szereplő leírások, azaz) ennyi és nem több alapján illik megértenie mindenkinek, aki valaha is kapcsolatba kerül a rekurzió kifejezéssel a jövőben:





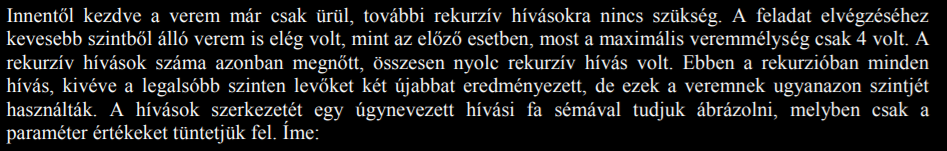
(további részletek a jegyzetben)

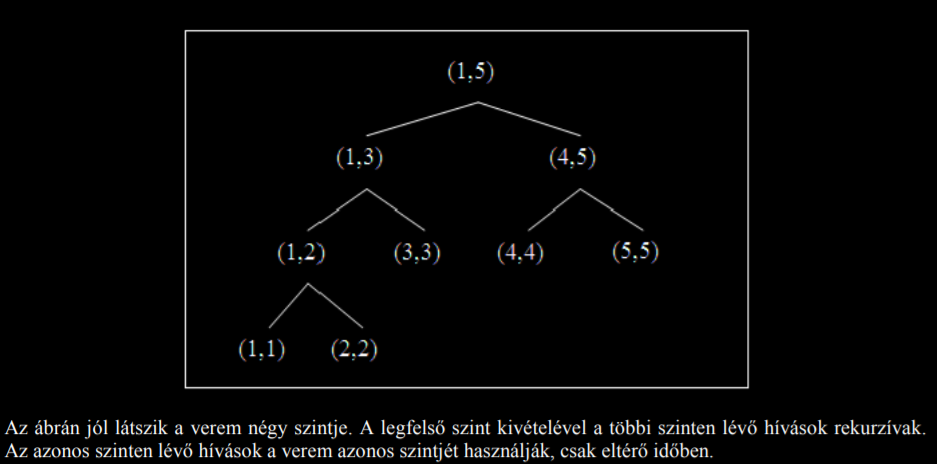
Ill.



(további részletek a jegyzetben)

Jelen cikk szempontjából releváns az alábbi idézet:

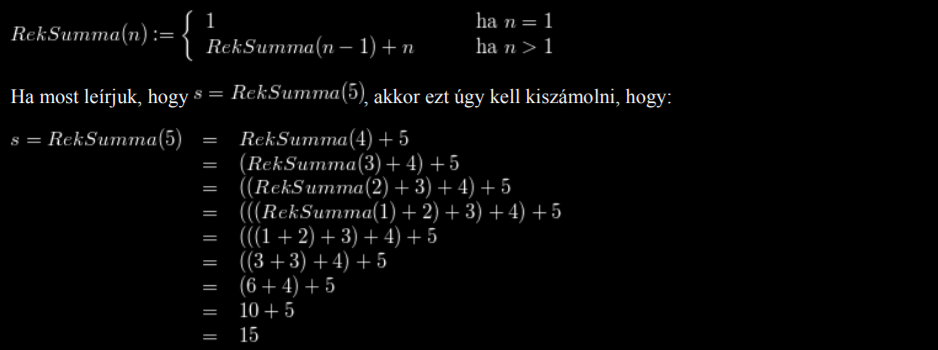


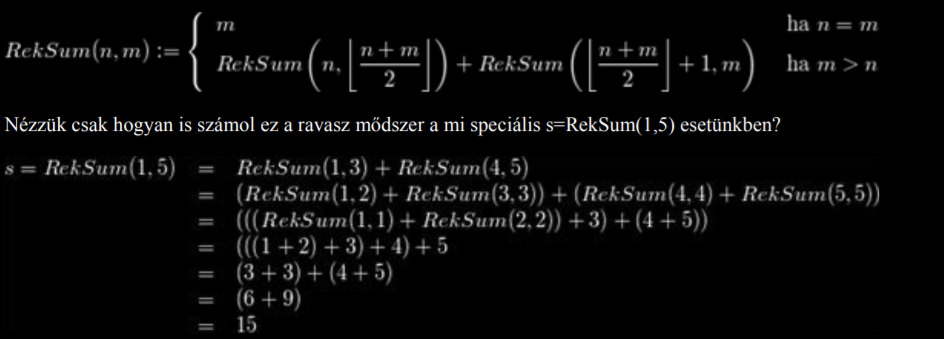


Az alternatív megoldások tehát ösztönös, világszemléleti alapon merülnek fel, ahol a mindenkori tudásszint és motivációs rétegek eredőjeként tekint egy-egy alternatívára valaki, mint saját ötletére. Az alternatívák (objektumok) értékelése egymáshoz képest olyan attribútumokat vár el, melyek minden alternatíva esetén értelmezhetők és melyeknek iránya van: pl.

* Minél kevesebb a veremszintek(mélység) száma, annál egyszerűbb (jobb) egy alternatíva.
* Minél kevesebb a rekurzív hívások száma, annál egyszerűbb (jobb) egy alternatíva.
* …

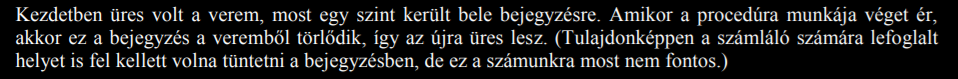
# 6. Példa – Nem csak a programozási nyelvek jelentenek kihívást





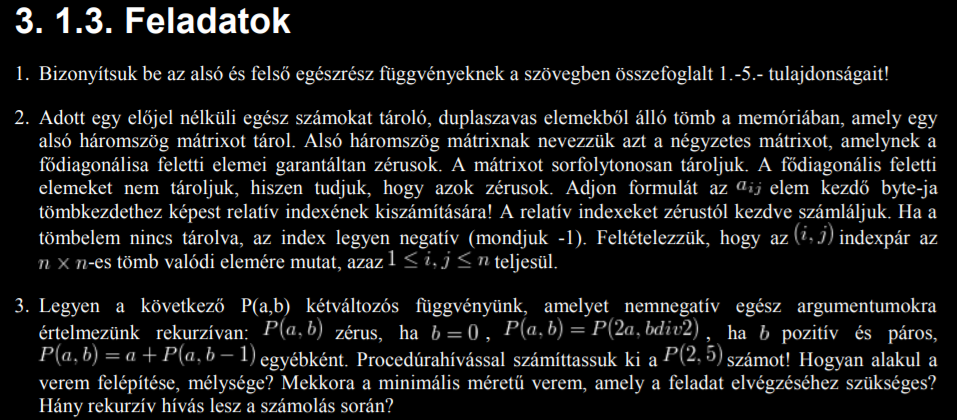
A programozási nyelvek mellett a feladatok leírásának többféle matematikai/egyéb módja is létezik (vö. folyamatábra).

Az alternatívák leírásakor érdemes figyelni vannak-e közös részletek, mint pl. a két rekurzív alternatíva esetén az alábbi sor: (((1+2)+3)+4)+5) – ami a rekurziók létjogosultságát is újra gondolni engedi a laikus megoldással szemben.



A fenti idézet zárójeles része pedig arra hívja fel a figyelmet, hogy a részletgazdagság foka is lehet önkényes adott pillanatban.

# 7. Példa – Ami egyáltalán nem vagy nem feltétlenül üzemmérnök-informatikus szint



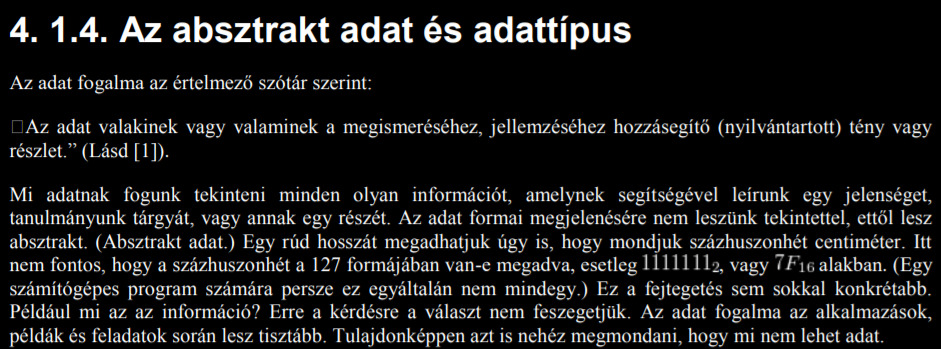
Az 1. feladat (bizonyítás) legyen elsődlegesen a matematikusok és/vagy az MSC-szintű képzettségre vágyók dolga – de ha szükséges egy bizonyítás, akkor ezt a szükségszerűséget fel kell ismerni, a bizonyítást meg kell tudni rendelni és a kapott eredményekkel tudni kell továbblépni. A bizonyítás helyességének ellenőrzése (vagyis a bizonyítási folyamat megértése – ennek bonyolultságától függően elvárható a BPROF-szintű kiképzendőktől is).

A 2. feladat: egy üzemmérnök informatikus számára CSAK valós élethelyzetekhez kötött feladatok illik, hogy létezzenek. Az absztrakciós szintet meg kell hagyni az MSC-diplomát szerezni kívánók számára.

A 3. feladat határeset: Mivel az alternatív megoldások összehasonlító értékelése kapcsán a verem mélysége és a rekurzív hívások száma attribútumként ismerhető el, így ezek levezetni tudása lehet adott esetben BPROF-szintű elvárás – de az sem szégyen egy üzemmérnök esetén, ha az attribútumok értékeit mással (MSC-szintű tudást hordozókkal) vezetteti le.

Ha bármely, a valós élethelyzetre utaló jelektől megfosztott feladat kapcsán a valós élethelyzetek legalább egyike újra rekonstruálható, akkor természetesen a teoretikus alapvetések azonnal értelmet nyernek az üzemmérnök informatikus képzés számára is.

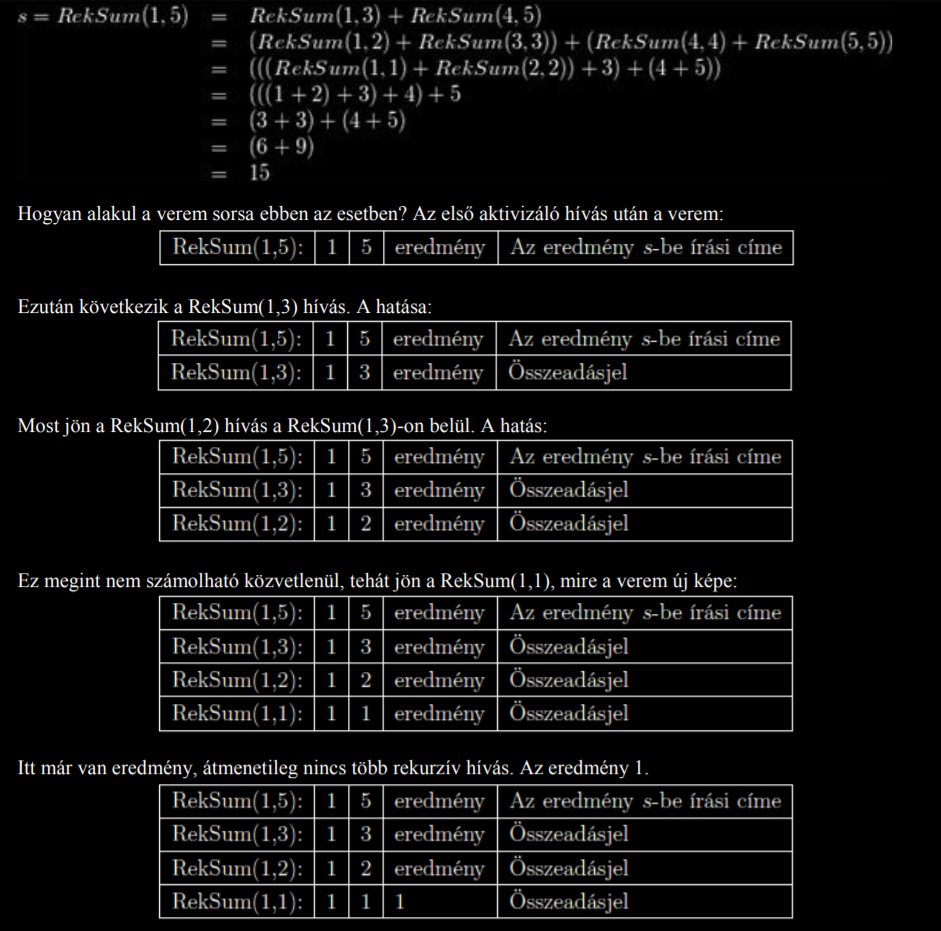
# 8. Példa – Ahol egyértelművé válik, mit nem lehet és nem kell tudni BPROF-szinten?!



Az adat-információ-tudás egymáshoz képesti lehatárolása az információs társadalom minden érintettjének meghaladni látszik komplexitáskezelő képességét. Jelen esetben felmerül az adat<információ értelmezés (vö. „adat … minden olyan információ”). Más megközelítésben minden adat, s valaki számára adott pillanatban hasznosan értelmezhető adat az információ (adat>információ), ahol az adat és az információ szavak, mint halmazjelölők értelmezendők és a relációs jel a halmazok elemszámának viszonyát jelzik (vö. jelezhetnék pl. az értékességet is, ahol vélhetően az adat<információ azt jelenti, hogy az adat általában értéktelenebb, mint az információ általában – sőt, az adattemetők fogalma mentén, az adat léte lehet károkozó = költséggeneráló is…)

Az információ jelenségének „nem feszegetése” ott, ahol a munkakört üzemmérnök informatikusnak és nem üzemmérnök „adatusnak” hívják, ahol információs többletérték, információs társadalom kulcsszavak köré publikációk szinte végtelen sora került már eddig is kidolgozásra, alapvetően nem megengedhető könnyítés.

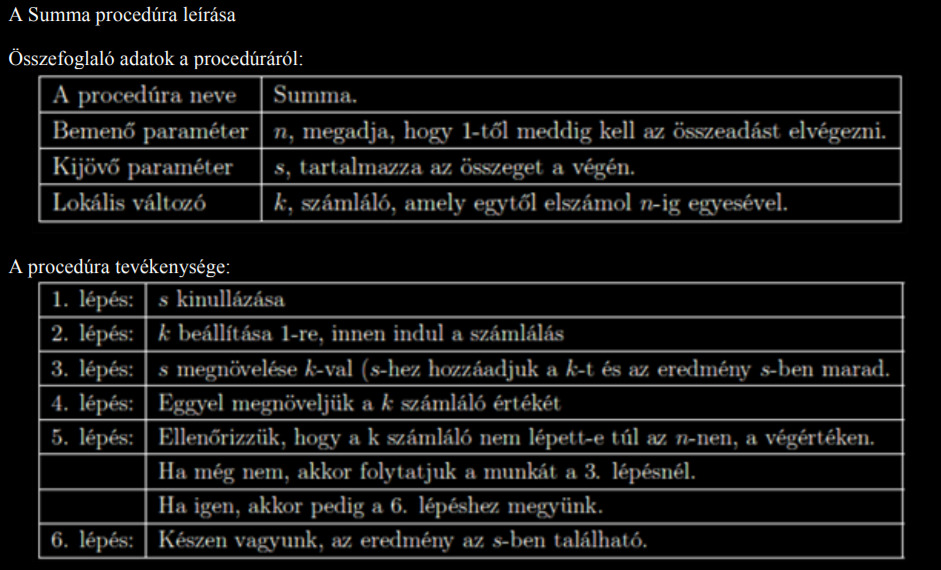
# 9. példa – A verem szekvenciális képei, avagy egy újabb blockchain kötődés?



A fenti részlet a vizsgált jegyzet 7. oldalán található, s folytatódik a 8. oldalon is legalábbi ilyen hosszúságban. Feltételezve, hogy egy BPROF képzés utolsó két féléve keretében a specializáció kulcsszava a blockchain, vajon a fenti tranzakció-jellegű sorozat értelmezhető blockchain-kihívásként is? Ha nem igazán érthető a kérdés, akkor az kettőt jelenthet: maga a témafelvetés is idegen a hétköznapi logikáktól, ami így akár innováció is lehetne, ha lenne egyébként értelme, vagy eleve nincs is értelme. Mivel az esélye annak, hogy egy speciális nézetet sikerülhet quasi véletlen tapogatódzással fellelni, talán még nem nulla, így talán érdemes precízebben megfogalmazni a kérdést: Tegyük fel, hogy valaki ír egy programot - pl. a 2009-es idegen jegyzet szerint az egész számok 1-től n-ig való összeadására, tegyük fel ismét csak, hogy valaki átveszi az irányítást a fenti programot futtató számítógép felett operációs rendszer szinten, ÉS semmi mást nem csinál, mint hogy belematat (quasi véletlenszerűen, de nem letálisan) a verembe, aminek az eredménye nem lefagyás, hanem esetlegesen téves végeredmény lesz, ez az esetleges tévesség a leggonoszabb dolog, ami egy felhasználóval/programozóval történhet, mert nem reprodukálható csak úgy. Ha azonban a verem jelenlegi kezelése helyett a veremképek blockchain-ként kerülnének tárolásra (vö. speciális compiler-rel), akkor garantálható lenne, hogy minden manipuláció (azaz már az első is biztonsági/hiba-jelzést eredményez vagyis a program vagy nem fut le a vészjelzés után, vagy vészjelzés esetén az eredményt nem tekintjük hitelesnek. Azt, hogy fizikailag ehhez a blockchain-értelmezéshez mit kell csinálni a megvalósulás érdekében – legyen egyelőre fekete doboz, de logikai szinten érdekesnek tűnhet az elvi lehetőség, hogy a programok futását ilyen módon is lehessen védeni - bár, ha még nem támadott senki ilyen módon, akkor ez persze teljesen feleslegesnek tűnik itt és most...

# 10. Példa - A verem-mélység és a rekurzív-hívásszám automatikus feltárása

A KNUTH-i elv, miszerint tudás az, ami forráskódba átírható, itt és most azt várja el, hogy egy-egy alternatív megoldásnak adott probléma esetén a verem-mélységre és/vagy a rekurziós hívásszámra vonatkozó értékét automatikusan, azaz algoritmus alapján lehessen levezetni. A 9. példa veremképei vajon elegendők/megfelelők egy ilyen algoritmus inputjaként?



A laikus megoldás kellően strukturáltan került leírásra ahhoz, hogy a veremmélység értéke meghatározható legyen automatikusan, ill. a rekurziós hívásszám értékéről be lehessen látni, hogy az pl. 0?

# 11. példa – Munkakörök/tudásszintek szétválasztása

A korábbiakban már említésre került, hogy 1-től n-ig a számok összegét képezni sokféle módon lehet, s nem informatikai kérdés az n\*(n+1)/2 alak ismerete. Hasonlóképpen nem informatika feladatok az alábbiak sem: <http://epa.oszk.hu/00200/00220/00043/pdf/firka_EPA00220_2005_2006_05_205-207.pdf>, noha ezek címe az informatikára utal. A szorzás (ill. később a gyökvonás) műveletének eltérő logikai menet szerinti elvégzése (vö. a vizsgált jegyzetben az orosz-paraszt-módszerre való utalás) csak annyiban informatikai kihívás, amennyiben a KNUTH-i elv eredményeként minden tudást előbb-utóbb forráskódba konvertálandó. A BSC/BPROF szint közötti különbségtételre ezen a ponton érdemes kitérni, mert eddig ez a két rövidítés a felsőfokon belüli alapszintet egyszerre jelölte. Mivel azonban a két jelölés léte nem véletlen, így a BSC szinttől - ízlés dolga, de – el lehet várni bármilyen életidegen(né tett) absztrakció értelmezését (vizsgahelyzetekben is), míg a BPROF kapcsán csak valós = üzemi körülmények értelmezése lehet a jegyadó feladat. Ha valaki megalkotja a legkülönfélébb szorzási (gyökvonási) logikai meneteket/lépéssorokat, akkor ennek programkódba való átültetése már lehet at BPROF-végzettségű személy feladata.

Így a vizsgált idegen jegyzet kapcsán pl. az alábbi szakkifejezések ismerte (tesztben való visszatükrözni tudása, különösen tetszőleges keresést támogató korpuszok használata mellett = valós önképzési helyzetet szimulálva) racionális, de még inkább racionális ezek valós élethelyzetekhez kötni tudása (vö bináris műveletek):

* Diszjunkció
* Konjunkció
* Antivalencia
* Ekvivalencia
* Implikáció
* Peirce-nyíl
* Scheffer-vonás, ill.
* Kettős tagadás
* Kommutativitás
* Asszociativitás
* Disztributivitás
* Idempotencia
* Konstansok hatása
* Elnyelés
* Ellentmondás
* Harmadik kizárása
* De Morgan, valamint
* Izomorfizmus
* Half-adder
* Full-adder
* Lemma, …

Ahhoz, hogy két inputjel eredményeként valamilyen output jel álljon elő, a fenti kifejezések bevezetése nem szükséges egy üzemmérnök informatikus számára.

Az alábbi idézet világosan jelzi, hogy a vizsgált jegyzet is (immár nem is először – vö. adat vs. információ) világos lehatárolásokat tesz arra vonatkozóan: mit tart adott ponton relevánsnak és mit nem. Ezen cikk BPROF-értelmezései arra tesznek kísérletet, hogy ezen határvonalat kifejezetten a BPROF-képzés gyakorlatiasságát védendő húzza meg, s természetesen mindenkor szabad lehetőséget adva az érdeklődőknek, hogy az LLL keretei között olyan irányba és mértékben törjenek ki a BPROF-ot jellemezni kívánó minimum-feltételek rendszeréből, amilyen iránya és mértékben csak akarnak.



# Lezáró gondolatok az 1. részhez

Ahogy azt az akkreditáció elnyerése kapcsán használni deklarált jegyzet első részeinek feldolgozása világos mutatja, a mindenkori Hallgató köteles szembesülni a kánon elvárásaival, amennyiben át kell, hogy olvassa a kötelező irodalmat, de egyben segítséget kap ahhoz, hogy ennek mely részei milyen módon dolgozandók fel. A kánon-orientált elvárások jelentős része a prompt megértésen túl nem válik (mert nem tud) gyakorlatias ismeretté. S a prompt megértést is a minden háttér-forrás felhasználása melletti tesztelés kell, hogy számon kérje „csak”. Ez a számonkérési mód az LLL egyik tesztje, vagyis a képesség arra, hogy a kiképzendő tetszőleges témakörökről racionális komplexitással tudjon kellően gyorsan nyilatkozni. De nem az a szint, amit quasi rutinszerű (felelősséggel tovább-szolgáltatandó) tudásként kell elvárni.

További részletek: <https://miau.my-x.hu/bprof>

1. A cikkben minden idézet inverz képként kerül megjelenítésre. [↑](#footnote-ref-1)