

TECHNIKÁK ÉS ESZKÖZÖK A TANULÓI FELADATOK ÉS DOLGOZATOK AUTOMATIZÁLT ÉRTÉKELÉSÉRE

A. Y. Montgomery

*Royal Melbourne Institute of Technology, Department of Computer Science
Melbourne, Australia*

A jelenleg hozzáférhető számítógép-alapú oktató- vagy tanulócsomagokban rendelkezésre álló tanulói értékelőrendszerekben a következő kérdés-típusokat találjuk:

- 1) egyszerű többszörös választás feladat azonnali válasz-visszacsatolással;
- 2) többszörös választás feladat, ahol a válasz az anyagban másutt szerepel: késleltetett visszacsatolás;
- 3) többszörös választás feladat több helyes válasszal: bármely helyes válasz teljes pontot ér;
- 4) igaz/hamis, ahol a rossz válasz pontlevonással jár — vagy járhat;
- 5) rövid választ igénylő kérdés három részből, ahol a válaszra utaló elemek a szövegben szerepelnek;
- 6) a tanulási célok köréből származó kulcsszó-listák elemeinek összepárosítása;
- 7) a diák munkájának leadása a számítógépes rendszeren keresztül, amelynek része a tanári értékelés, valamint hiperlinkek a világhálózathoz, levelező- és newsgroupokhoz, amelyek révén a helyes válasz megtalálható;
- 8) rövid választ igénylő, egy audio-file-ban megadott anyagra alapozott kérdések;
- 9) számítási feladatok minden tanuló számára különböző számértékekkel;
- 10) videó-anyaghoz kötődő többválasztós feladat;
- 11) rövid ábramagyarázatot kívánó feladat;
- 12) többszörös választás feladat cím és ábra alapján;
- 13) randomizált szövegváltozókkal megadott szakterminusok és definíciók párosítása;
- 14) rövid választ igénylő kérdés egy másik felhasználáshoz vezető linkkel.

E felsorolt 14 kérdéstípus közül azok (különösen az 1., 2., 3., 4., 6., 10., 12. és 13. típus), amelyekre több válasz lehetséges, leginkább a diákok alapvető fogalmi ismereteit vizsgálják, illetve egyszerű problémákat elemző készségeiket. E kérdéstípusok előnye az, hogy könnyen specifikálhatók és az oktató/tanuló szoftver azonnal automatikusan értékelheti őket. Ezáltal segíti a visszajelzést, illetve a diákok tanulását is. Az egyes diákok-

nak feltett kérdéseket a feladatbankokból véletlenszerűen ki lehet választani oly módon, hogy az adott oktató- és tanulómodul specifikus céljainak teljesítését értékelje, s így nagyobb esélyt adjon arra, hogy a válaszok az egyes diákoktól, ne pedig valamely „csapat-teljesítményből” származzanak. Nem tagadjuk, hogy ezek az alapvető tényanyag tanulá-sában hasznos segítséget nyújtanak a diákoknak. Ha a tanároknak olyan oktató/tanulócso-maguk van, amely nem tartalmazza e módokat, olyan termékeket kell választaniuk, ame-lyek felkínálják ezeket.

A mérnöki és más műszaki kurzusok célja azonban az, hogy fejlesszék a diákok ké-pességét egyre bonyolultabb vagy nyitott végű problémák elemzésére, a problémák meg-oldásainak szintetizálására (gyakran „design” formájában), vagy a megoldás természetes nyelven való kifejtésére. A fogalmak pusztá ismerete és az egyszerű problémák elemzé-sére való képesség nem elegendő e kritikus, magasabb rendű képességek fejlesztésére és a jelenleg rendelkezésre álló oktató/tanulócso-magok lehetőségei nem adnak módot ezek automatikus értékelésére.

A 7. típusú válaszok értékelése nem azonnali és nem automatikus. Az értékelő oktató később ad visszacsatolást, amely egyénre szabott tanácsokat tartalmaz és motiváló hatása is lehet a megszólított diákra. Megszenvedti ugyan az azonnaliság hiányát, de jelenleg ez a nyitott végű vagy a design-típusú problémák uralkodó értékelési módja. A másik lehe-tőség programozási feladatok esetében az, hogy a programokat egy tesztelő és végrehajtó rendszeren futtatják, amely lefordítja a programot. A tesztadatokat prioritás szerint állítja sorba és így értékeli. A tesztkörnyezetet úgy kell szervezni, hogy ha a program végre-hajtása leállna végrehajtási hiba miatt, a kontroll visszakerüljön az értékelőkörnyezethez és ne a teljes értékelési kontrollrendszer abortáljon!

A 8., 11. és 14. típusú kérdések esetében a megengedhető válaszok köre általában egyetlen kifejezésre vagy mondatra korlátozódik. A diák vagy megadja a „kijelölt vá-laszt”, vagy nem: sem a szürke zóna, sem a „Jó irányba halad, de nem gondol erre és er-re” típusú tanács nem lehetséges ebben a rendszerben.

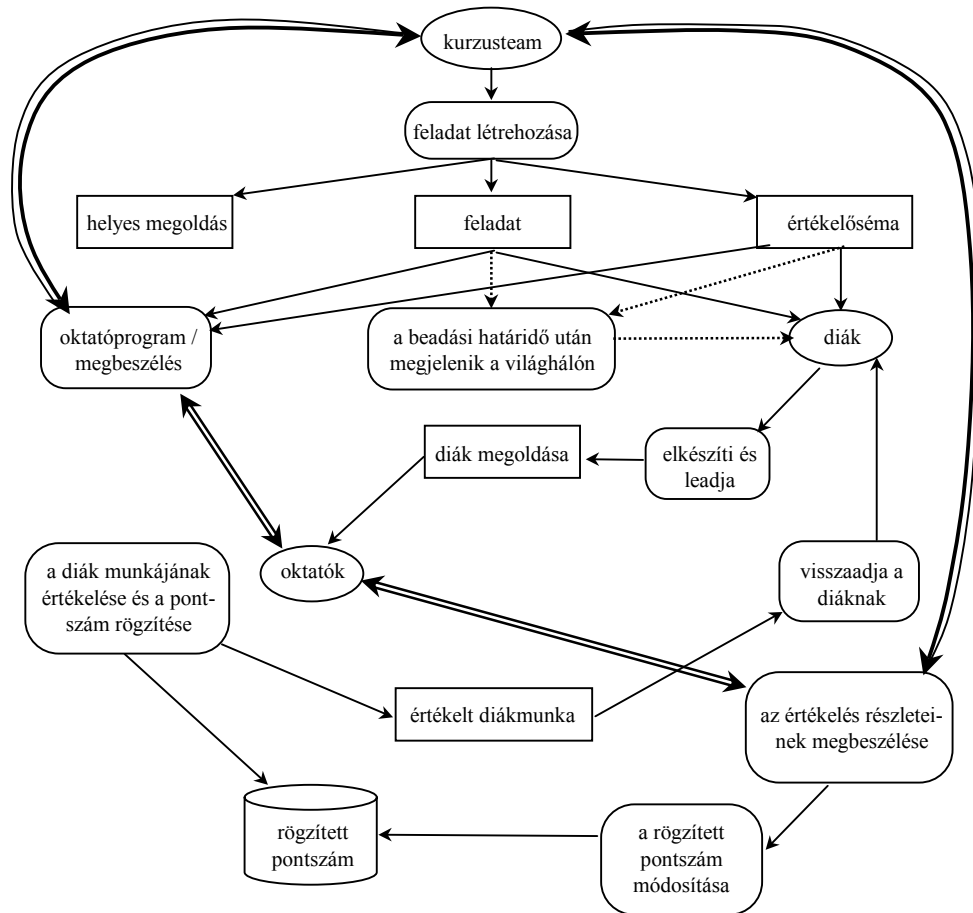
Az az elérendő cél, hogy a diákok ne csak a fogalmakat ismerjék, illetve ne csak egy-szerű problémákat tudjanak elemezni, hanem képesek legyenek technikai problémák megoldásainak szintetizálására és komplex (szakmai) fogalmak kifejezésére is. Vannak, akik szerint ezeket a duális képességeket – a komplex helyzetek elemzését, és azok meg-oldásainak szintézisét, akárcsak a komplex helyzetek megoldásának világos megfogal-mazását – minden egyetemet végzettben magas szintre kell fejleszteni.

Jelen munkánk a diákoktól e magasabb szintű készségek működtetését kívánó fela-datokra és vizsgakérdésekre várt válaszok automatizált értékelésének újabb fejleményeit állítja a középpontba (ld. 1. és 2. ábra)

A numerikus számításokat kívánó designfeladatok értékelését segítő eszközök

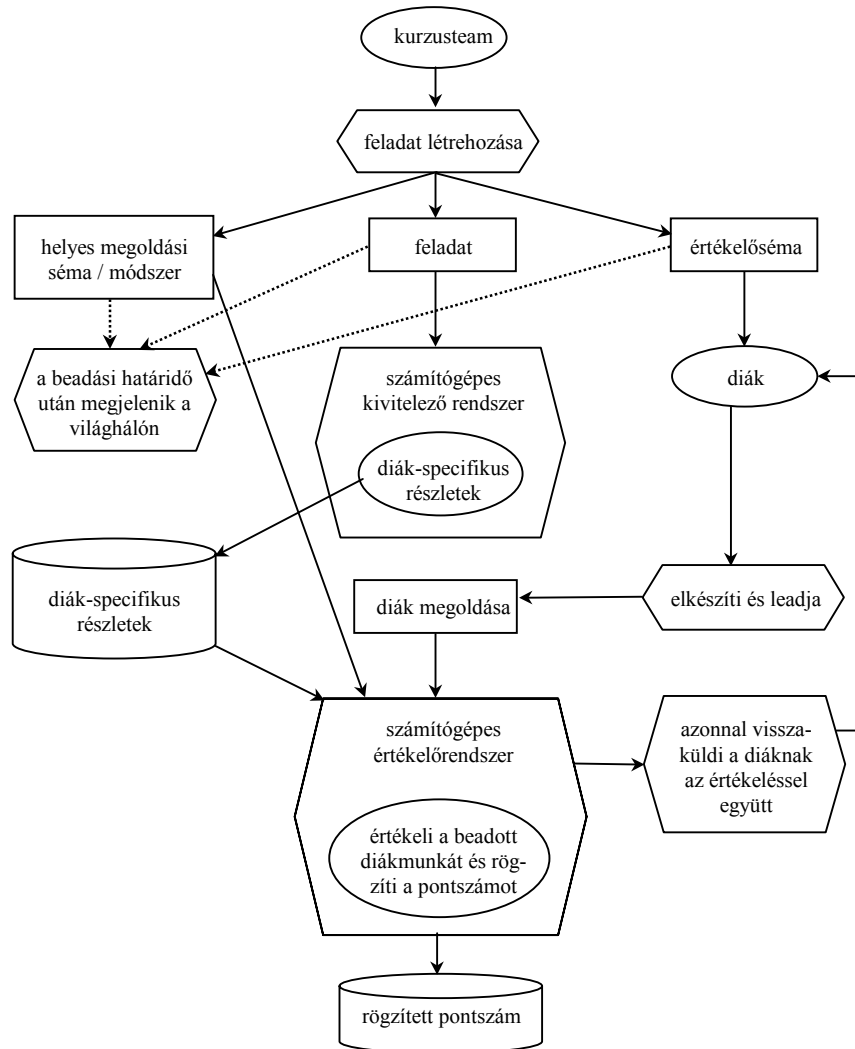
A fentebb felsorolt többi 13 kérdéstípussal szemben a 9. teret ad a diákoknak arra, hogy design- vagy megoldás-szintetizáló képességét megmutassa. Ám az eddig rendelkezésre álló eszközök általában e lehetőséget a komputációs/matematikai/numerikus tarto-mányra korlátozták. A függelékben bemutatott (1) példa egy ilyen rendszer (szándéko-san) leegyszerűsített használatát mutatja be. Magam e kérdéstípust kitűnőnek találtam

még nagy, többszetevős, komplex komputációs design problémák esetében is, ezért melegen ajánlom ezt a rendszert műszaki kari kollégáimnak, amennyiben még nem találkoztak volna vele.



1. ábra
A jelenlegi rendszer

A diákok beadott anyagainak oktatói értékelése — Hosszú üresjáratok az értékelés elkészítésében — Az inkonzisztenciák vitához vezetnek — A tevékenységek, amelyekre a legtöbb idő fordítható: az oktatók oktatása és az egyes diákok értékelésének 'egyeztetése'.



2. ábra
A kívánt rendszer

A feladatmegoldások kiszámítása diákspecifikus paraméterek használatával — A megoldások értékelése „megengedhető tolerancián” belül — A diákok eredményének értékelése fuzzy logika felhasználásával — A tevékenységek, amelyekre a legtöbb idő fordítható: a feladatok finomítása, az értékelő megjegyzések illetve a diákok munkáját értékelő és számukra visszacsatolást adó modell finomítása.

A nyíltvégű kérdések értékelésére hivatott rendszer kívánatos jellemzői

Mind a teljes értékű, mind a nem teljesen jó válaszok megfogalmazása kitűnő képességek birtoklását kívánja meg a kurzusfejlesztő teamtől: a terület szakértőjétől, azaz a területen jártas szaktanártól (már feltételezve, hogy a gyakori hibák ismertek és leírtak) és egy megbízható, absztrakt szimbólumokkal foglalkozó szakértőtől.

A vizsga- és a nyitott végű feladatok automatizált értékelésére szolgáló hasznos rendszernek a következő tulajdonságokkal kell rendelkeznie:

- a) felismer és teljes pontszámmal értékeli minden egyes helyes választ (vagy azok halmazát), és megfelelő, teljes pontszámot ad.
- b) felismer minden egyes részben helyes választ (vagy azok halmazát), a válasznak megfelelő irányító megjegyzéseket tesz, illetve a beadott válasz helyességével arányos pontszámot ad.

A helyes válasz specifikálásához rendelkezésre álló módszerek

A tudásterületek széles körében lehetséges a teljesen helyes válaszok kifejezése a következő formák egyikével:

- 1) Betűlánc(ok) halmaza, amely betűről betűre összevethető a diák beadott megoldásával.
- 2) Egyszerű template, amely egy, például *Perl* (Wall és *mtsai*, 1996) nyelven írt „script” révén használható a feladat helyes megoldását jelentő sorozat (vagy sorozatok halmazának) összevetésére.

Egy ilyen script képes például a betűláncok esetében a nagybetű/kisbetű megkülönböztetés; figyelmen kívül hagyására elválasztó karakterek (pl. szóköz, sor-emelés) zajkarakterek (pl. indokolatlan *, #, @ stb. a kurzusteam döntése alapján) mellőzésére; arra, hogy ne zavarja meg a lezáró szimbólumok hiánya (pl. “)”) vagy “;” a program kontextusában); hogy megengedje ekvivalens fogalmak alternatív karakterekkel való jelölését (pl. elfogadja az „x” jelet a „*” helyett); hogy figyelmen kívül hagyja az egyes szövegszerkesztők által beillesztett formázási vagy fontspecifikációs karaktereket és így tovább, amennyiben a kurzusteam megfelelőnek tekint ilyen műveleteket.

- 3) Jelentéssel bíró tokenekből (a token terminus szinonimái a kontextustól függően az ikon, szimbólumsor stb.) álló formális készlet formális grammatikával, amely a tokenek összes megengedhető kombinációját meghatározza. Ilyen formális készlet és nyelvtan specifikálható kontextusfüggetlen grammatikákra (a LALR-ra és alacsonyabb rendű formátumokra) Lex, Yacc és más hasonló nyelveken (ilyen lexikai elemző és összevető összeállító-összeállító rendszerre lásd például *Aho* és *Ullman*, 1977).
- 4) A fenti képességek támogatását olyan eszközök adhatják, amelyek – felismerik a szótöveket (az előtagok és szuffikszumok leválasztásával ott, ahol ezek elrejtik a szemantikailag szignifikáns összetevőt) és

– lehetővé teszik szinonimák használatát (szinonimaszótárak, szópárosító képesség stb. révén) olyan helyzetek kezelésére, ahol a diákok válaszaiban a várt terminusok helyett szemantikailag majdnem egyenértékű kifejezések szerepelnek.

- 5) Összetettebbek azok a képességek, amelyek megkívánják, hogy a kurzusteam meghatározza a várt helyes válasz szemantikáját, például a Conceptual Graph [fogalmi gráf] (Sowa, 1984) technika felhasználásával. A CG szemantikai elemző-csomagba kerülő input, például Peirce (Ellis, 1994), egy természetes nyelvi elemzőcsomag outputja, mely utóbbi azonosítja a releváns nyelvi tokeneket és szemantikai kapcsolataikat.

A legtöbb kontextusban azonban sokkal nehezebb megvalósítani a (b) képességet.

A helytelen válaszok specifikációjához rendelkezésre álló módszerek

Nem könnyű például a (b) pont kívánalmait kielégíteni még a számítógépes programozási nyelveken készült diákmunkák esetében sem. Az ilyen nyelveknek szükségszerűen vannak formális grammatikai szabályai, amelyek lehetővé teszik a helyes megnyilvánulások felismerését. A számítógépes nyelvek formális grammatikáinak felhasználásával a program forrásszövege lefordítható a számítógép által végrehajtható formába. Az érvénytelen megnyilvánulások felismerésére, a hiba pontos jellegének és a programon belüli pontos helyének azonosítására azonban csak magasan fejlett nyelvspecifikus fordítóprogramok hatékonyak. E nehézség okai túlságosan is a fordítóprogram-technológia körébe tartoznak, ezért itt most nem foglalkozunk velük. Ám e hiányosságok jelentik a fő akadályt, ami miatt nem használhatjuk őket saját céljainkra.

E nehézségek általános hatása az, hogy míg egy formális grammatika elég gyakran létrehozható és felhasználható a diákok válaszaiknak összevetésére a helyes megoldásokkal, ez nem áll olyan megoldások esetében, ahol a válaszok helyessége tág határok között változhat és ahol a nem teljesen helyes válaszokra releváns, hasznos megjegyzések tehetők.

Úgy tűnik, a következő a legjobb megvalósítható megoldás (ld. a 3. példát):

- A tanár definiál a feladatra adható minden részlegesen helyes válaszváltozatot és az ezekre adható pontértéket;
- Mindegyik válaszhoz definiál egy tokent a lexikai elemzőben való használatra;
- E tokenek specifikációjának sorozata a leghelyesebbtől a legkevésbé megfelelő, de még elfogadható válaszig terjed;
- A „szövegszkenner” először a leghelyesebb megoldást próbálja megkeresni a diák munkájában;
- Amennyiben nem talál meg egy adott megnyilatkozást, a szövegszkenner „visszaállítja” az input szöveg keresőpontját a diák munkájában az adott feladat kezdetére és új keresést indít, amely a sorozatban következő tokenre irányul;
- Amennyiben valamely tokent megtalál, a szövegszkenner beiktat egy hasznos hibakommentárt tartalmazó jelet és egy pontszámot. Például „E feladatban nem megengedett a kapcsolási műveletek használata”; „1 pontot ért el a lehetséges 10-ből”;

- Amennyiben nem találja meg a definiált tokenek egyikét sem, a szövegszkenner hiba-kommentárt helyez el, például „Nem jó megoldás” és „0 pontot ért el a lehetséges 10-ből”;

Mindezek alapján úgy tűnik, hogy egy fordító-fordító jellegű eszköz használata feleslegesen túlbonyolítja az ilyen rendszer implementálását éppúgy, mint valamely formális grammatika definiálása. Ez a megközelítés azonban megkívánja, hogy a diák munkájának szkennelése minden definiált részlegesen helyes válaszra megtörténjék.

„Ha valamit érdemes csinálni, akkor érdemes azt akár rosszul is csinálni”

Lehetnek-e egyszerűbb módszerek egy adott feladat automatizált értékelésére még akkor is, ha ezek nem adnak tökéletes oktatói segítséget? Egy ilyen rendszer létjogosultságát az igazolhatja, ha „azonnali visszacsatolást ad, mégpedig nem sokkal gyengébb minőségűt, mint az oktatók többsége, akiktől viszont csak három héttel később vagy azon is túl érkezik visszajelzés”, továbbá ha „a leadott munkának csak egyetlen szkennelése szükséges az értékeléshez”.

Tekintsük át a 2. és 4. példában bemutatott megközelítést. Ehhez nincs szükség komplex grammatikára vagy fordító-fordító technológiára, hanem a folyamat a következőképpen épül fel:

- 1) A kurzusteam meghatároz minden elfogadható választ, a társuló hibaüzeneteket és a megfelelő pontszámot;
- 2) Az így azonosított „lehetséges válaszok” elemzésével részlegesen helyes tokeneket definiálnak és mindegyik tokenhez egy súlyt rendelnek;
- 3) Perl scriptet készítenek, amely értékeli a „lehetséges válaszokat”. Megszámol minden token típust, ezt beszorozza a tokenenkénti súllyal, és összegzi a szorzatokat minden token típusra majd megállapítja minden beadott munka összpontszámát;
- 4) A diákok munkáját ugyanezen Perl script pontozórendszerrel értékelik: a pontozóprogram minden munkát balról jobbra leolvass, minden előforduló tokenet megjegyyez és összeadja a token súlyának megfelelő pontszámokat minden előfordulásakor;
- 5) A diáknak adott hibaüzenet és jegy az lesz, ami a diák munkájával összegyűjtött pontszámhoz legközelebb lévő „megengedhető megoldáshoz” tartozik;
- 6) Néhány esetben a legközelebbi „megengedhető megoldás” az lesz, ami a „Nem jó megoldás”-hoz tartozik s aminek az eredménye „10-ből 1 pontot ért el”;
- 7) Ha kívánatos és lehetséges, külön ellenőrzés végezhető a beadott munka grammatikai helyességének megállapítására és a munka e jellemzőjére kiegészítő pontszám adható.

A megközelítés alapja

A módszer feltételezi, hogy az érvényes megoldások körén kívül eső válaszok olyan pontszámokat eredményeznek, amelyek inkább a nullához, semmint más értékhez állnak

közelebb és így inkább jogosan használt "Nem jó megoldás" oktatói megjegyzést és ahhoz tartozó pontszámot eredményeznek. Abban a néhány esetben, amikor a diák irreleváns oktatói észrevételt kap, a rendszer a kielégítő szintnél gyengébb lesz. Azt, hogy az esetek milyen arányban esnek az irreleváns oktatói megjegyzések kategóriájába, az jelöli ki, hogy a kurzusteam milyen gonddal határozza meg a megengedhető válaszokat, azonosítja a megengedhető tokeneket és súlyozza ezeket.

Összefoglalás

Jelen munkánkban áttekintettük a diákok nyitott végű feladatokra készült munkái értékelésére rendelkezésre álló vagy fejlesztés alatt lévő lehetőségeket.

Első pillantásra a helyes válaszok formális nyelvtan vagy formális fogalmi gráf formájában történő specifikálása ígéretesnek tűnik. Az ilyen specifikációk automatikusan generált lexikai és szintaktikai elemzőprogramokhoz vezethetnek. Annak a kívánalomnak az elismerése azonban, hogy hasznos oktatói észrevételeket adjunk a diákoknak, a figyelmet ráirányítja a csupán a helyes válaszokat felismerő módszerekről azokra, amelyek a maximálisból megfelelő arányú pontszámmal értékelhető különböző helyességi fokú válaszokra összpontosítanak. A rendkívül szokatlan alkalmazások kivételével a helytelen válaszokat nehéz formális grammatika vagy formális fogalmi gráf formájában definiálni.

Ebben a kontextusban úgy tűnik, hogy a különböző helyességi fokú válaszok listájának összeállításával és egy egyszerű szövegmegfeleltető és értékelőrendszer alkalmazásával jó egyensúly teremthető a kurzusteam ráfordított energiája, az oktatói hatékonyság és a komputációs egyszerűség között.

Fordította: Molnár Edit Katalin

Irodalom

- Aho, A. V. és Ullman, J. D. (1977): *Principles of compiler design*. Addison Wesley, Reading, USA.
- Ellis, G. (1996, szerk.): Case studies in conceptual graphs. Technical report 94/2. RMIT, Department of Computer Science, Melbourne.
- Sowa, J. (1984): *Conceptual structures: Information processing in mind and machine*. Addison Wesley, Reading, USA.
- Wall, T., Christiansen, T. és Schwartz, R. L. (1996): *Programming Perl*. 2. kiad. O'Reilly & Associates.

ABSTRACT

A. Y. MONTGOMERY: TECHNIQUES AND TOOLS FOR THE AUTOMATED ASSESSMENT OF
STUDENT ASSIGNMENTS AND EXAMINATION SCRIPTS

This paper discusses methods for assessing student submissions for assignments or examinations which require responses to „open-ended questions”. Such responses require higher level skills than concept awareness and simple problem analysis. For such questions students must synthesise solutions, recommendations or designs. The assessment methods discussed draw upon established Computer Science tools and techniques or ones still under development.

Key Words & Concepts

Automatic assessment; student assignments & examination scripts; open-ended questions and designs; rapid assessment; intelligent assessment

Magyar Pedagógia, **100**. Number 4. 459–472. (2000)

Levelezési cím / Address for correspondence: Royal Melbourne Institute of Technology,
School of Computer Science and Information Technology, GPO Boks 2476 v, Melbourne,
3001, Victoria, Australia

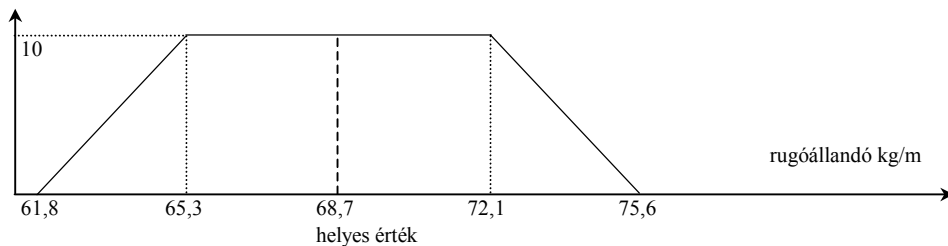
Függelék

1. példa

Adja meg a k rugóállandót (kg/m), amely m tömeg hatására x hosszúságot nyúlik!

paraméter	kívánt mértékegység	terjedelem	értékválasztás/ pontosság	megengedett/ maximális tűrés
megnyúlás x	cm	10–30	véletlenszerű / a legközelebbi 0,1-re kerekítve	
tömeg m	kg	2–15	véletlenszerű / a legközelebbi 0,1-re kerekítve	
rugóállandó k	kg/m	a diák megoldása	a diák megoldása	$\pm 5\%$ / $\pm 10\%$

paraméter	a diák által megadott érték	a terjedelemben eső legalsó/legfelső érték	a maximális tűrés legalsó/legfelső értéke
megnyúlás x cm	14,7		
tömeg m kg	10,1		
rugóállandó kg/m	a helyes megoldás 68,7	72,1 / 75,6	65,3 / 61,8



3. ábra
Fuzzy értékelőséma

paraméter	a diák által megadott érték	a terjedelemben eső legalsó/legfelső érték	a maximális tűrés legalsó/legfelső értéke
megnyúlás x cm	14,7		
tömeg m kg	10,1		
rugóállandó kg/m	a helyes megoldás 68,7	72,1 / 75,6	65,3 / 61,8
a diák válasza	6,9 kg/m	a diák pontszáma: 0/10	

Megjegyzés a diák számára: „Az Ön megoldása 6,9 kg/m.
A helyes megoldás 68,7 kg/m. Az Ön válasza kívül esik a +/- 10%-os tűréshatáron.
A pontszáma 10-ből 0.”

2. példa

FELADAT: Határozza meg a relációs adatbázis-kezelő rendszerek kontextusában a szuperkulcs és a kulcs fogalmát a reláció szempontjából.

Helyes meghatározás

Egy reláció szuperkulcsa egy attribúció-készlet, amelynek értékei egyedi módon meghatároznak egy n elemből álló rendezett halmazt. A kulcs minimális szuperkulcs. A szuperkulcs-készletből nem távolítható el attribúció anélkül, hogy ez ne szüntetné meg az egyedi identifikációs tulajdonságot.

A szuperkulcs meghatározása

kulcsszó vagy kifejezés	elfogadható szinonima	súly
szuperkulcs	szuper kulcs	1
reláció	relációs séma	1
attribúció-készlet	attribúció halmaz	1
egyedi		2
értékek		5
identifikál	specifikál, definiál, meghatároz	1
-n elemű rendezett halmaz	sorozat	1
összpontszám		12

A kulcs meghatározása

kulcsszó vagy kifejezés	elfogadható szinonima	súly
kulcs		1
minimális	minimum	1
szuperkulcs	szuper kulcs	1
attribúciók		1
készlet		1
megszüntet	eltávolít	3
eltávolít		1
egyedi		1
identifikáció		1
képesség	jellemző	1
összpontszám		12

A diák beadott válasza:

„A szuperkulcsok olyan attribútum, amelyek egy n elemű rendezett halmazokat identifikálnak. A kulcs lehet elsődleges vagy nem elsődleges és minimális.”

Szuperkulcs

kulcsszó vagy kifejezés	pontszám
szuperkulcs	1
attribúciók	1
identifikál	1
n elemű rendezett halmaz	1
összpontszám	4/12

Szintaxis: a szuperkulcs definícióban

szintaxishiba	pontszám
alany és állítmány egyeztetése	-1
határozatlan névelő többes számú főnév előtt	-1
összes szintaxis pontszám	-2
összpontszám	2/12

A diák beadott válasza: kulcs

kulcsszó vagy kifejezés	pontszám
kulcs	1
minimális	1
szuperkulcs	
attribúciók	
készlet	
megszüntet	
eltávolít	
identifikál	
képesség	
elsődleges: irreleváns	-1
nem elsődleges: irreleváns	-1
összpontszám	0/12

Szintaxis: a szuperkulcs definícióban

szintaxishiba	pontszám
a kijelentés szemantikailag nem egyértelmű	-1
szintaxis pontszám	-1
összpontszám	0/12

3. példa

A MOVIES adatbázishoz (itt nem mellékeljük) írjon egy SQL lekérdezést, amely létrehozza azon rendezők felsorolását, akik nem rendeztek egyet sem az adatbázisban szereplő filmek közül (5 pont).

Helyes megoldás

```
a) SELECT dirname
   FROM Director
   WHERE dirnumb NOT IN
   (SELECT dirnumb
    FROM Movies);
```

```
b) SELECT dirname
   FROM Director
   WHERE NOT EXISTS
   (SELECT dirnumb
    FROM Movie
   WHERE Director.dirnumb = Movie.dirnumb);
```

Helytelen megoldás

```
SELECT dirname
FROM Director
WHERE EXISTS
  (Select dirnumb
   From Movie
  WHERE Direcor.dirnumb <> Movie.dirnumb);
```

Először számba vesz minden lehetséges választ amelyért pont adható, beleértve a gyakori hibás megoldásokat is. Az összehasonlító program először figyelmen kívül hagyja az üres közöket (szóközök, tabulátorok, soremelések), minden változónevet következetes, elfogadható formára alakít, majd az első tételtől kezdve keresi a táblázat tételeinek megfelelőit. Ha nem találja az első tétel megfelelőjét, a következőkben megkísérli a diák munkáját a táblázatban tárolt sorozat minden tételének megfeleltetni. Ha sikerül az azonosítás, a következő inputkaraktereket a beazonosítottak utáni helyről veszi. A beazonosítás sikertelensége visszaállítja a keresőpontot az azonosítandó rész elejére.

Az ehhez hasonló komplex mintázatok és/vagy parciális mintázatok könnyű azonosítását segítheti, ha a következő lezáró tokenekkel formális grammatikát definiálunk.

Lezáró szimbólum-szekvencia (megengedhető token)	Pont	Megjegyzés	Összes elért pontszám 5-ből
A) SELECT dirname FROM Director	1	Az indítás jó	
B. WHERE dirnumb NOT IN	1	Jól halad	
C. SELECT dirnumb FROM Movie);	3	Remek munka, teljesen helyes válasz	5
D. SELECT dirnumb FROM Movie	1,5	Hiányzó lezáró jelek	3,5
E. WHERE NOT EXISTS	1	Jó irányban halad	
F. SELECT dirnumb FROM Movie	1	Eddig jó	
G. WHERE Director.dirnumb = Movie.Dirnumb);	2	Remek munka, teljesen helyes válasz	5
H. WHERE Director.dirnumb = Movie.Dirnumb	0,5	Hiányzó lezáró jelek	3,5

A. Y. Montgomery

Lezáró szimbólum-szekvencia (megengedhető token)	Pont	Megjegyzés	Összes elért pontszám 5-ből
I. WHERE Director.dirnumb <> Movie.Dirnumb);	0	Egysoros output lesz az eredmény minden helyzetben, ahol a rendező NEM a választott film rendezője. Ezt kéri a feladat?	1
#1		A megoldás további része rossz	0
#2			2
#3			3
#4			1

A ponttal értékelhető megoldás lehetséges grammatikája
 $[A \#1], [\{ B, (C \mid D \mid \#2) \} \mid \{ E, F (G \mid H \mid I \mid \#3) \} \mid \#4]$

Megjegyzések a grammatikához:

- | opcionális (VAGY-gyal összekötött vagy diszjunktív) szekvenciát jelez
- , karakterlánc kapcsolódást jelez, amely ténylegesen kötelezően bekövetkező (ÉS-sel összekötött vagy konjunktív) szekvencia
- #1,2,3,4 négy különböző hibahelyzetet jelöl

4. példa

Specifikálja a leggyakrabban használt reprodukálható módszereket egy szoftverfejlesztési projekt költségvetésének és időbeosztásának megbecsülésére.

A módszer rövidítése	A módszer teljes megnevezése	Kapható pontszám	A diák válasza	A diák pontszáma
COCOMO	Constructive Cost Model	2	Constructed Costing Model {Azonosítás a „Construct — Cost Model”szótóval}	1
FP	Function Point	2	IFPUG (Function Point) {Azonosítás az „FP”-vel és a Function Point”-tal is}	2
SLIM	Putnam-Myer	2	HR Buildup {Releváns fogalom, de nem azonosítható a definiált válaszokkal }	0
	Feature Points	1	Features {Szótó azonosítása mint „Feature”}	1
	Analogy	1	Analogous {Helyesírási hibával}	0
	ÖSSZESEN	8	A DIÁK ÖSSZPONTSZÁMA	4