

## Методи оцінювання семантично різноманітної відповіді у системі комп'ютерного тестування знань

Здебільшого у комп'ютерному тестуванні використовується методика, що базується на так званому методі множинного вибору (multiple choice). Головною позитивною рисою цієї методики є простота комп'ютерної реалізації. «Платити» за цю простоту доводиться педагогу–конструкторові тесту. Саме він відповідає за ретельність добору тестових питань, здатних адекватно оцінити знання учня. І, незважаючи на наявність різноманітних методичних рекомендацій щодо такого добору, якість тесту більшою мірою залежить від педагогічного мистецтва його упорядника.

Та найбільш важливим є те, що навіть «досконалим» тестам типу multiple choice притаманні серйозні психолого-педагогічні вади. Насамперед, це істотне обмеження самостійної діяльності учня. Якщо при звичайному опитуванні учень сам повинен формулювати свої думки, то у ситуації множинного вибору він лише вибирає відповідь із запропонованих готових варіантів. Думаємо, що наявне надто широке використання згаданої методики не є культурно та психологічно виправданим, бо послаблення вимог до самостійності мислення учня не сприяє його повноцінному розвитку.

Оцінка знань учня, виставлена комп'ютером, заслуговувала б значно більшої довіри, якби обмеження на можливу форму учнівської відповіді були якомога мінімальними. Тому розширення можливостей застосування у системах комп'ютерного тестування знань відповідей «відкритої форми» має бути, з нашого погляду, пріоритетним напрямком розвитку таких систем. На жаль, на сьогодні не можна говорити ані про наявність значного числа відповідних розробок, ані про різноманітність рішень, які пропонуються. Сучасні системи, що допускають відкриті відповіді, по суті обмежуються традиційним методом «ключових слів».

У цій роботі ми пропонуємо розглянути можливості, які надаються для вирішення проблеми оцінювання відкритих відповідей системою комп'ютерного тестування Control. Ці можливості, на нашу думку, є значно ширшими, ніж у методу ключових слів.

У системі Control для ідентифікації текстової відповіді застосований алгоритм, який дозволяє комп'ютеру розпізнавати вірні і невірні відповіді, ігноруючи несуттєві граматичні та синтаксичні помилки і не звертаючи уваги на стилістику відповіді доти, доки зберігається розпізнаваність її смислу [5]. Цей алгоритм ґрунтується на розроблених його авторами математичних методах обчислення чисельних мір подібності символічних послідовностей [3; 4], застосованих до оцінки подібності слів і фраз натуральної мови. Завдяки цьому комп'ютер може задавати учням питання у тій самій формі, яка вживається у підручнику, задачнику чи під час усного опитування.

Наприклад, у сеансі комп'ютерного тестування учнів 7 класу з теми «Інформаційне забезпечення комп'ютера» курсу інформатики (див., наприклад, [2, с. 51-57]) можна задати у точності ті ж контрольні питання, які наведено у кінці відповідного розділу підручника [2]. При цьому зрозуміло, що запитання: «Як вказують тип файлу?» [2, с. 57] потребує відкритої форми відповіді. Фактичні відповіді різних учнів на це запитання можуть збігатися з наведеним у підручнику «еталоном»:

«Тип файлу записують відразу після імені і відокремлюють від імені крапкою»

(див. [2, с. 51]); але можуть бути і такими (тут і далі вважаємо, що граматичні та стилістичні особливості відповідей належать учням; слова з помилками, гіпотетично допущеними учнем, виділяємо курсивом):

- 1) «Записують після імені і *видокремлюють* крапкою»;
- 2) «Тип файлу вказують після крапки, що стоїть за іменем»;

- 3) «Після імені ставлять крапку та дописують тип»;
- 4) «Через точку після імені»;
- 5) «За ім'ям файла, відокремивши точкою»;
- 6) «Записавши до трьох символів після точки в кінці імені файла»;
- 7) «Дописавши його до імені файла після крапки»;

і таке інше.

Завдяки наявності означеного алгоритму, всі ці фрази можуть бути визнані правильними відповідями на поставлене питання. Водночас невірними відповідями слід було б визнати:

- 8) «Перед іменем файла, відокремивши точкою»;
- 9) «Одразу після імені файла»;
- 10) «За точкою»;

тощо. Неважко помітити, що такого поділу відповідей на вірні та невірні можна досягти, вибравши за ключові слова «кінець», «ім'я» та «крапка». Однак, на відміну від традиційного методу, застосований у системі Control алгоритм подібності дозволяє вживати ці слова у різних відмінках, допускати у них незначні помилки, користатися синонімами, тощо.

Використання розробленого алгоритму можна зробити більш ефективним, якщо врахувати специфіку ідентифікації відповідей неелементарної структури. Вивчення структурних особливостей тексту - необхідна умова створення і налагодження дієвого механізму автоматизованої оцінки складної відповіді.

В даній роботі ми хочемо звернутися до розгляду проблеми оцінки семантично різнорідних відповідей, тобто таких відповідей, які можливо розділити на достатньо самостійні в смисловому відношенні елементи.

В подальшому ми будемо використовувати терміни з теорії комп'ютерного тестування знань, визначені зокрема в [5] та в [4], такі як *дескриптор* і *правило висновку*. Дескриптором називається описувач допустимих відповідей на задане питання, який зберігається в базі даних системи тестування. Правило висновку – це алгоритм порівняння дескриптора з відповіддю, що надійшла від учня, при цьому виходом алгоритму є деяка оцінка відповіді.

Наприклад, для наведеного вище тестового питання дескриптор, що оцінить відповіді 1) – 7) як правильні, а відповіді 8) – 10) як невірні, має наступний вигляд:

```
R      {0.4:за\пи\сують^вказують^ставлять} & {після^за^кінці^закінчення}
R      0.4:&{\ім\ені^назви} {відокремлюють^стоїть^ставлять} & {крапк^точк}
RPR_Fr #0.6,0.35,21
```

Розрізняються *сингулярні* (такі, що вживаються незалежно від інших) і *комбіновані* правила висновку. Прикладом сингулярного правила є використане у наведеному вище дескрипторі так зване «правило RPR\_Fr» призначене для порівняння текстів методом подібності фраз. У роботі [5] докладно викладається алгоритм роботи правила RPR\_Fr, і вводяться поняття *міри подібності* слів та фраз, *глобального* та *локального рівнів* подібності.

Комбіноване правило дозволяє застосувати для оцінки однієї відповіді кілька сингулярних правил. Це необхідно, зокрема, задля врахування ймовірності отримання різного типу правильних відповідей. Наприклад, на запитання «Яка довжина діагоналі квадрату зі стороною 1?» можна дати як текстову відповідь: «квадратний корінь з 2»; так і числову: 1.41, або ж 1.4142 тощо. Кожна з таких відповідей оцінюється специфічним саме для неї сингулярним правилом висновку, але щоб допустити обидві відповіді, слід звернутися до комбінованого правила. В даному випадку можна застосувати так зване правило RPR\_0 [4], що дозволяє будувати загальну оцінку відповіді за допомогою математичних та логічних операцій. Для розгляданого тестового питання дескриптор

правила RPR\_0 буде являти собою, спрощено кажучи, формулу **A OR B**, де **A** і **B** – оцінки, які були виставлені сингулярними правилами, а **OR** – операція диз'юнкції. Зауважимо, що у цьому випадку як **A**, так і **B** є оцінками відповіді, узяті «в цілому».

Суть нашої роботи зводиться до побудови нового комбінованого правила висновку, яке б дозволяло оцінювати різноманітними сингулярними правилами *семантично різноманітні частини* відповіді.

Навіщо необхідно виділяти у відповіді окремі смислові блоки, і як це може допомогти при оцінюванні правильності відповіді? Ми вкажемо три основні причини доцільності саме такого підходу.

Перша причина пов'язана з тим, що зростання розміру текстового блоку, який ми бажаємо ідентифікувати за допомогою деякого сингулярного правила, веде до ускладнення задачі формування адекватного дескриптора. Для «довгих» текстів виникає занадто багато смислових нюансів і різноманітних способів висловлювання, які необхідно врахувати, щоб охопити всю безліч правильних відповідей. При збільшенні розміру текстового блоку рано чи пізно ми стикаємось з межею, поза якою ефективність сингулярного правила стає нижче допустимої. Отже, текст необхідно розділити на більш дрібні, але відносно самостійні частини.

Друга причина: оцінка повинна бути диференційованою, повинна мати додаткові градації, окрім «вірно» чи «невірно». Відповідь може бути не зовсім вірною або частково вірною. Коли кажуть, що відповідь частково правильна, це означає, що якась її частина правильна, а інша – ні. Отже, треба вміти знаходити ці частини у відповіді.

І, нарешті, третя причина: існування різноманітних способів висловлення одної думки. Коли ми висловлюємо ту саму думку, ми можемо вибирати різні, але близькі за смислом слова, а також по-різному будувати фразу. Сингулярні правила висновку можуть врахувати ймовірність заміни того чи іншого слова, однак стикаються з труднощами у випадку переформулювання висловлювання з суттєвою зміною його структури. В цьому випадку ідентифікація може бути успішною, якщо ми виділяємо структурні частини фрази та вказуємо їх альтернативні формулювання.

Отже, актуальною є проблема розпізнавання комп'ютером різних смислових частин відповіді. Перейдемо до опису алгоритмів, здатних, на наш погляд, розв'язати ряд задач, пов'язаних з цією проблемою. Про алгоритми ми будемо говорити, наводячи приклади їх застосування до конкретних тестових питань, які є можливими в сеансах комп'ютерного тестування.

Почнемо з наступного запитання (з курсу інформатики для учнів 7 класу):

«Чому не можна використовувати оперативний запам'ятовуючий пристрій для тривалого зберігання програм і даних?» [2, с.34]

Відповідь у підручнику сформульовано так:

«При вимиканні живлення комп'ютера вміст оперативного запам'ятовуючого пристрою втрачається.» [2, с.32]

У тексті відповіді можна виділити наступні смислові частини:

- 1) при вимиканні живлення комп'ютера;
- 2) вміст оперативного запам'ятовуючого пристрою втрачається.

Для ідентифікації цих частин ми використаємо ключові слова, які ідентифікують контекст; в подальшому такі слова називаємо *ключами 1-го порядку*.

Можна сказати, що ключ 1-го порядку для укладача дескриптора грає роль певної смислової «квінтесенції» словесного вислову. Скажімо, якими б словами не передавати смисл наведеної вище частини 2), навряд чи можливо обійтись без слова «втрачається» або його контекстних синонімів («стирається», тощо).

Для наведеної «еталонної» відповіді можна (у «першому наближенні») вважати ключами 1-го порядку, специфічними для кожної з її двох частин, відповідно:

- 1) вимиканні;
- 2) втрачається.

Ключі 1-го порядку повинні мати властивості обов'язковості, тобто без них відповідна частина буде завідомо невірною, та унікальності, тобто вони не повинні зустрічатися в іншому контексті.

Після того, як ми ідентифікували одну з частин, необхідно обмежити її, тобто виявити її границі.

З цією метою вводяться *ключі 2-го порядку*. Спрощено кажучи, це слова, які за смыслом розгляданої фрази належать до «оточення» ключа 1 порядку, і є визначальними у цьому «оточенні». Хоча між ключами 2-го порядку у реченні можуть стояти інші (*не* ключові) слова, ключі 2-го порядку утворюють особливий «ланцюжок», який починається на початку розгляданої смислової частини речення, а закінчується у її кінці. Трохи нижче ми дамо точні визначення; а поки зазначимо, що в нашому випадку такі «ланцюжки» складатимуть слова:

- 1) «при, живлення, комп'ютера» – для першої частини; та
- 2) «вміст, пристрою» – для другої.

Для більш детального пояснення нам знадобиться спочатку ввести визначення поняття *відстані* між словами в тексті. Під відстанню між словами ми будемо розуміти кількість ключових слів (враховуючи ключі як 1-го, так і 2-го порядку), що знаходяться між двома даними словами. Позначимо через  $r(a, b)$  відстань відповідно між словами  $a$  та  $b$ . Тепер введемо поняття *зв'язаності* двох слів тексту за заданою множиною  $A$  слів. Будемо говорити, що слова  $a$  і  $b$  зв'язані, якщо знайдеться послідовність  $\alpha_1, \alpha_2, \dots, \alpha_n$  слів з множини  $A$ , таких, що

$$r(a, \alpha_1) = r(\alpha_i, \alpha_{i+1}) = r(\alpha_n, b) = 0.$$

Перейдемо до опису алгоритму виділення частини відповіді. Передусім, у відповіді знаходиться ключ 1-го порядку тієї частини відповіді, яка має найбільшу вагу (є найважливішою частиною). Після цього, починаючи з ключа 1-го порядку, шукаються всі зв'язані ключі 2-го порядку за множиною ключів, які відносяться до цієї частини. Початок та кінець побудованого ланцюжка і будемо вважати границями даної частини.

Проілюструємо сказане на прикладі пошуку частин у такій учнівській відповіді, яка цілком збіглася з наведеною вище еталонною: «При вимиканні живлення комп'ютера вміст оперативного запам'ятовуючого пристрою втрачається.» Нехай частинам еталонної відповіді призначено вагу: відповідно першій – 0.25, другій – 0.75. Отже, частина на 2) є вагомішою, і аналіз відповіді учня починається з пошуку ключа 1-го порядку 2-ої частини – слова «втрачається». Зв'язаний з ним ключ 2-го порядку – «пристрою». З ключем «пристрою» зв'язаний також ключ «вміст», бо між словами «вміст» та «пристрою» інших ключів нема. Отже, границі 2-ої частини визначаються словами «вміст» і «втрачається». Всі слова в цьому проміжку зв'язані за множиною ключів 2-го порядку для 2-ої частини. Аналогічно знаходимо, що слова «при» і « комп'ютера» задають границі 1-ої частини відповіді.

Кожен з ключів, як 1-го, так і 2-го роду, може мати один або декілька синонімів. Тому необхідно задати групу близьких за смыслом чи за функцією у реченні слів-ключів, одне з яких обов'язково повинно зустрітися в правильній відповіді. Наприклад маємо такі синоніми: «пристрою», «ОЗП» та «пам'яті».

Нерідко буває, що семантично пов'язані між собою слова однієї частини відповіді «потрапляють усередину» іншої його частини. Людина (носій мови) легко розрізняє частини і в цьому випадку. Наприклад, наступна відповідь цілком вірна:

«Тому, що вміст оперативного запам'ятовуючого пристрою  
при вимиканні живлення комп'ютера втрачається».

Для адекватної оцінки комп'ютером цієї відповіді потрібно «скасувати» вимогу щодо порядку ідентифікації частин відповіді за спаданням їх ваги. Справді, за цієї вимоги при

встановлених значеннях ваги спочатку буде виділятися 2-а частина, її границі будуть визначені невірно, і відповідь отримає незадовільну оцінку. Тому для тих випадків, коли одна з частин відповіді може входити в іншу (скажімо, 1-а частина може входити у 2-у, і виділення частин необхідно здійснювати, починаючи з 1-ї), передбачений особливий параметр R\_ORD, в якому вказується бажаний порядок виділення частин.

В нашому випадку задамо: R\_ORD 1,2.

Щоб читач міг наочно уявити собі роботу системи Control, ми будемо супроводжувати розглядані варіанти тестових питань імовірними учнівськими відповідями на них і тією оцінкою, яку система поставить кожній з таких відповідей.

Зокрема, остаточний варіант дескриптора нашого нового правила, складений для щойно розглянутого тестового питання з урахуванням усіх зазначених параметрів, дає такі результати (за 5-бальною шкалою)<sup>1</sup>:

«Вимикання живлення комп'ютера *призведе* до втрати вмісту пристрою»  
– Ця відповідь отримає оцінку 5.

«Вміст ОЗП при вимиканні живлення комп'ютера втрачається» – 5.

«Оперативний запам'ятовуючий пристрій не можна використовувати для тривалого зберігання програм і даних тому, що вимикання живлення комп'ютера веде до втрати вмісту цього пристрою» – 5.

«Бо його вміст втрачається при вимиканні» – 4.

«Після виключення пам'ять стирається» – 4.

«При вимиканні ми втратимо вміст оперативної пам'яті» – 4.

«Тому, що вміст втрачається» – 3.

«Вміст пристрою можна втратити» – 3.

«Бо ОЗП вимикається» – 2.

Наступне тестове завдання, яке ми розглянемо, відноситься до курсу «Алгебра та початки аналізу» [1, с.316]:

«Сформулюйте означення похідної функції в точці,  
не використовуючи символічні позначення.  
Похідною функції в точці називається...».

Поставлене у такій формі, запитання вимагає відповіді натуральною мовою. (Воно могло б бути задане учневі на усному іспиті. Комп'ютерний варіант потребує застереження про небажаність вживання математичних символів). Еталонна відповідь з підручника звучить так:

«границя відношення приросту функції до приросту аргументу  
за умови, що приріст аргументу прямує до нуля, а границя існує»

[1, с.295]<sup>2</sup>.

<sup>1</sup> Сам текст дескриптора тут не наводимо. Трохи нижче ми подамо приклади повних текстів дескрипторів для інших тестових задач. З цих прикладів читач зможе отримати уявлення про загальну структуру та розміри дескрипторів розгляданого нами правила.

<sup>2</sup> Ми внесли у визначення з підручника невеликі зміни: усунули позначення  $\Delta x$ ,  $\Delta y$ ,  $y=f(x)$ ,  $x=x_0$ . Зміни зроблено з огляду на вимогу не вживати у відповіді символічні позначення.

З метою спрощення подальшого викладу будемо вважати уточнення «а границя існує» несуттєвим, тобто приймемо, що його присутність чи відсутність на оцінку відповіді учня не впливає (неважко побудувати дескриптор, який враховуватиме це уточнення; проте принципівих відмінностей від поданого нижче він не матиме).

Виділяємо смислові частини:

- 1) границя відношення приросту функції до приросту аргументу;
- 2) приріст аргументу прямує до нуля.

Ключі 1-го порядку: 1) функції; 2) нуля.

Ключі 2-го порядку: 1) границя, відношення, приросту, аргументу; 2) приріст, прямує.

Наступну учнівську відповідь можна визнати стилістично не дуже вдалою, але цілком вірною фактично:

«Границя (при прямуванні приросту аргументу до нуля)  
відношення приросту функції до приросту аргументу».

Тому порядок визначення частин у відповіді учня задамо так: R\_ORD 2,1.

У дескрипторі обговорюваного нового комбінованого правила (назвемо його, з метою наступних посилань, RPR\_2 ) ключі 1-го порядку записуються в рядку з назвою R\_par1, а 2-го порядку – R\_par2. Розділювачем ключів є кома. Так само, як при укладанні дескриптора для правила RPR\_fr, разом із задаванням ключів доречно визначати глобальні і локальні рівні подібності слів у фразі.

Наведемо дескриптор, побудований для оцінки правилом RPR\_2 відповіді на розглядане питання. Читач може не робити детальний «розбір» цього дескриптора – для розуміння статті такий розбір є необов'язковим. Дескриптор наведений головним чином для того, щоб читач зміг оцінити його розмір і спосіб зображення.

```
RPR_2 границя відношення приросту функції до приросту аргументу
      за умови, що приріст аргументу прямує до нуля
R_ORD 2,1
R_par1 $0.8$функції
R_par2 $0.6$границя, {відношення^частки},0.5% приросту
R_par2 %0.7% {аргументу^змінної}
R_par1 $0.6$ {нуля^0}
R_par2 $0.7%0.5% {приріст^останнє^знаменник},%0.4%прямує
0.6*
R ~# &0.6:приріст {відношення^частки від делення}
R & {0.6:приріст 0.6:функції} {до^поділений на^на} приросту
RPR_fr {аргументу^незалежної змінної} #0.7,0.72,22
+0.4*
R ~# & {приріст аргументу^0.5:останнє^знаменник} 0.4:\прям\ує до
RPR_fr & {0.6:нуля^0} #0.7,0,21
End_2.
```

Окрім еталонної відповіді, цей дескриптор визнає цілком правильними (оцінить на відмінно), зокрема, такі відповіді:

«Границя відношення приросту функції до приросту незалежної змінної,  
при прямуванні останнього до нуля».

«Границя частки від ділення приросту функції на приріст аргументу,  
при прямуванні останнього до нуля».

«Границя відношення приросту функції до приросту її аргументу, коли знаменник прямує до нуля».

А от приклади відповідей, яким система виставить інші оцінки:

«Границя відношення приросту функції до приросту незалежної змінної, якщо останнє наближається до нуля». – Відповідь отримає оцінку 4.

«Границя відношення приросту функції до приросту аргументу, коли обидва прирости прямують до нуля». – 3.

«Границя відношення приросту функції до приросту аргументу, коли приріст функції прямує до нуля». – 3.

«Границя відношення приросту аргументу до приросту функції». – 2.

«Границя приросту аргументу поділеного на приріст функції, коли приріст функції прямує до нуля». – 2.

«Границя приросту аргументу, коли приріст функції прямує до нуля». – 2.

Розглянемо ще один приклад тестового питання – з того ж підручника Алгебри за 10-11-тий класи.

Завдання:

«Що називається функцією?

У відповіді аргумент позначайте через  $x$ , а функцію – через  $y$  »

[1, с.76]

Еталонна відповідь:

«Залежність змінної  $y$  від змінної  $x$  називається функцією, якщо кожному значенню  $x$  відповідає єдине значення  $y$  »

[1, с.3].

Виділяємо частини:

- 1) залежність змінної  $y$  від змінної  $x$ ;
- 2) кожному значенню  $x$  відповідає єдине значення  $y$ .

Перша частина відповіді, взагалі кажучи, не дуже значуща, але нам вона здається бажаною з точки зору закінченості і повноти відповіді. В будь-якому випадку, незалежно від оцінки вагомості цієї частини, її наявність необхідно передбачити при укладанні дескриптора, бо скоріше за все учень буде намагатися навести визначення з підручника в цілому. Якщо у дескрипторі призначити першій частині вагу 0.2, а другій – 0.8, то за умови відсутності у відповіді учня частини 1) і повній правильності частини 2) він отримає оцінку «добре».

Ключі 1-го порядку: 1) залежність; 2) єдине.

Ключі 2-го порядку: 1) змінної,  $x$ ,  $y$ ; 2) кожному, значенню, змінної,  $x$ ,  $y$ .

У дескрипторі, окрім іншого, потрібно врахувати імовірність використання для позначення змінних як латиниці, так і кирилиці (щодо курсиву, то його вживання чи відсутність не впливає на оцінку, яку ставить правило RPR\_2).

Можна побудувати досить простий дескриптор, що оцінить відповідь:

«Залежність  $y$  від  $x$ , при якій кожному значенню незалежної змінної відповідає тільки одне значення залежної».

на «відмінно». Однак абсолютно правильною є також відповідь, що сформульована у такий спосіб:

«Залежність  $y$  від  $x$ , при якій різним  $y$  не можуть відповідати однакові  $x$ ».

У цій відповіді друга частина висловлена зовсім інакше, ніж у поданому вище еталоні. Щоб її врахувати, простіше за все долучити до дескриптора альтернативний варіант формулювання другої частини. Зробити це можливо завдяки наявності в системі Control засобів обчислення широкого класу логіко-арифметичних виразів. Замість формули:

$$0.2 * A + 0.8 * B,$$

де **A** і **B** означають відповідно 1-шу та 2-гу частини відповіді, ми використаємо формулу:

$$0.2 * A + 0.8 * (B \text{ OR } C),$$

де **C** відповідатиме новій альтернативі другої частини. Наведену правильну відповідь з альтернативним формулюванням частини 2) вкрай важко було б оцінити позитивно, використовуючи одне лише сингулярне правило RPR\_Fr.

Ще одна проблема постає перед нами у зв'язку з оцінкою відповіді:

«Залежність, де кожному значенню  $x$  відповідає одне значення  $y$ ».

Якщо виділяти в цій відповіді першу частину – «Залежність  $y$  від  $x$ », починаючи з ключа 1-го порядку «залежність», то знайдені ключі 2-го порядку цієї частини – « $x$ » та « $y$ » спричинять виникнення такої ситуації, коли вся відповідь буде включена до складу саме першої частини. З іншого боку, якщо встановити інший порядок виділення частин, при якому спочатку у відповіді буде виділятися друга частина, то, наприклад, у зразковій відповіді слова « $x$ » та « $y$ » з її першої частини будуть віднесені до другої, що знову призведе до невірної визначення границь частин відповіді.

Задля вирішення проблеми перетинання частин за множинами ключів 2-го порядку (як маємо у даному випадку), до складу дескриптора правила RPR\_2 включено ще один параметр – *недозволене слово*. Це таке слово відповіді, яке не може бути присутнім у межах чергової частини, що виділяється. У нашому прикладі недозволеним словом будемо вважати слово «кожному» – приймемо, що воно може з'явитись у другій, але не у першій частині відповіді. У дескрипторі правила RPR\_2 недозволене слово записується у рядку з назвою R\_par3. Якщо параметр R\_par3 задано для деякої  $i$ -тої частини відповіді, алгоритм працюватиме так: після знаходження ключа 1-го порядку  $i$ -тої частини шукаються пов'язані ключі 2-го порядку, але якщо на цьому етапі при просуванні вправо чи вліво від ключа 1-го порядку після чергового ключа 2-го порядку буде знайдено недозволене слово, тоді слово, яке передує недозволеному, буде вважатися останнім лівим або правим словом у межах  $i$ -тої частини. Наприклад, у відповіді:

«Залежність, де кожному значенню  $x$  відповідає одне значення  $y$ »

справа від ключа 1-го порядку «залежність» першої частини відповіді знаходиться недозволене слово «кожному», яке передує ключам 2-го порядку цієї частини – « $x$ » та « $y$ ». Тому ці ключі вже не будуть включені до складу 1-ї частини, отже першою частиною буде вважатися: «Залежність, де». Коротко кажучи, знайдене недозволене слово припиняє розширення відповідно правої або лівої границі частини відповіді.

Дещо спрощений варіант дескриптора для запитання, яке ми розглядаємо, має такий вигляд:



```

R_ORD 1,3,2
R_par1 $0.5$залежні\сть\
R_par2 $0.6${x^x^незалежної},{y^y^залежної}
R_par3 $0.6${кожному^будь-якому^одному^різним^відмінним}
R_par1 $0.6${єдине^одне^1}
R_par2 $0.6${кожному^будь-якому},значенню,змінної,{x^x^незалежної}
R_par2 {y^y^залежної}
R_par1 $0.6${різним^відмінним}
R_par2 $0.6${x^x^незалежної},{y^y^залежної},значенню,змінної,{однакові^самі}
0.2*
R ~# &0.5:залежність змінної &{y^y^залежної} від &{x^x^незалежної}
RPR_fr #0.6,0,22
+0.8*(
R ~# {кожному^будь-якому} значенню
R &{змінної x^змінної x^незалежної змінної^x^x}
R відповідає &{0.7:єдине^0.7:одне^1}
R значення &{y^y^залежної} змінної
RPR_fr #0.6,0.81,22
or
R ~# різним &{y^y^залежної змінної}
R &не {0.4:можуть^повинні} відповідають
R {однакові^ті самі^одне й те ж} &{x^x^незалежної}
RPR_fr #0.6,0.81,22
)

```

У результаті відповідь:

«Залежність, де кожному значенню  $x$  відповідає одне значення  $y$ »,

отримає оцінку «добре».

Приклади відповідей на розглядане питання та їх оцінок, отриманих правилом RPR\_2 (з трохи складнішим за наведений дескриптором):

«Залежність  $y$  від  $x$ , при якій різним значенням залежної змінної не може відповідати одне й те ж значення незалежної» – 5.

«Така залежність, при якій будь-якому значенню  $x$  відповідає єдине значення  $y$ » – 4.

«Різним  $y$  не можуть відповідати однакові  $x$ » – 4.

Ця відповідь, звичайно, не є «цілком» досконалою, проте свідчить про певні знання, тому оцінка 4, на наш погляд, є виправданою.

«Залежність  $y$  від  $x$ , при якій різним значенням залежної змінної може відповідати одне й те ж значення незалежної» – 2.

«Залежність  $y$  від  $x$ , при якій різним  $x$  не можуть відповідати однакові  $y$ » – 2.

Як можна помітити, серед множини оцінок відсутня оцінка 3. Це пов'язано з тим, що відповідь поділена на дві частини, яким призначені ваги відповідно першій – 0,2, а другій – 0,8. У такому разі, якщо у відповіді присутні обидві частини, кожна з яких вірна, відповідь отримує оцінку 5, якщо присутня хоча б лише друга частина і вона вірна,

відповідь отримує оцінку 4, якщо ж друга частина відсутня чи невірна, відповідь отримує загальну оцінку 2 (незалежно від того, як буде оцінено її першу частину). Для того, щоб зробити можливою оцінку 3, потрібно обидва альтернативні варіанти другої частини відповіді у свою чергу ще раз поділити на частини. При цьому, наприклад, обов'язкові слова «кожному» та «єдине» знаходяться у різних частинах, а тому відсутність лише одного з них не призведе до негативної оцінки всієї другої частини відповіді. Отже, матимемо, наприклад, для відповіді:

«Залежність, де значенню  $x$  відповідає одне значення  $y$ »

загальну оцінку 3. Однак обговорення питання “ієрархічного” поділу відповіді на частини (тобто питання подальшого поділу частин на частини), виходить за межі даної статті. Тому у наведеному вище прикладі ми обмежилися лише трьома оцінками.

Сукупність розглянутих прикладів оцінювання учнівських відповідей, на наш погляд, досить переконливо свідчить про ефективність використаних у системі Control методів такого оцінювання. Гадаємо, що у перспективі ці або подібні методи можуть повністю (чи майже повністю) замінити традиційну методику множинного вибору – і, таким чином, зробити автоматизований контроль знань подібним до «звичайного» людського.

## Література

1. Алгебра і початки аналізу: Підруч. для 10-11 кл. загальноосвіт. навч. закладів / М.І.Шкіль, З.І.Слепкань, О.С.Дубинчук. – 2-ге вид. – К.: Зодіак-ЕКО, 2000. – 608 с.
2. Жалдак М.І., Морзе Н.В. Інформатика-7. Експериментальний навчальний посібник для учнів 7 класу загальноосвітньої школи. – К.: «ДіаСофт», 2000. – 208 с.
3. Леоненко Л.Л., Поддубный Г.В. Новые принципы идентификации ответов и их реализация в автоматизированной системе тестирования знаний Control // Материалы научно-методической конференции «Программно-технические средства информатизации образования». – К., 1995. – С. 91-92.
4. Леоненко Л.Л., Поддубный Г.В. Теория подобия конечных последовательностей и ее приложение к распознаванию образов // Автоматика и телемеханика. – М., 1996. – №8. – С. 119-131.
5. Поддубный Г.В. Интеллектуальные системы и их использование в компьютерных технологиях обучения (на примере автоматизированной системы тестирования знаний Control). Методическое пособие с элементами лабораторного практикума. – Одесса, 1997. – 28 с.