Online assessments with parametric questions and automatic corrections: an improvement for MCTest using Google Forms and Sheets

Francisco de Assis Zampirolli[*], **Valério Ramos Batista,**
Edson Alex Arrázola Iriarte, Irineu Antunes Junior

[1]Federal University of ABC (UFABC)
Av. dos Estados, 5001 – Santo André – 09210-580 – SP – Brazil

{fzampirolli,valerio.batista,edson.iriarte,irineu.antunes}@ufabc.edu.br

Abstract. *In many areas of knowledge it has always been a challenge to evaluate students efficiently. Considering that we are all undergoing a pandemic period, efficient evaluations are necessary and urgent. In our paper we followed the main objective of adapting* MCTest. *Namely, a web platform devoted to generate and correct individualized exams automatically. We have addressed the problem of distance student evaluation by profiting* MCTest. *As a result it provides a solution that is free of charge and enables creating parametric questions with LaTeX and Python. The automatic correction is carried out with Google Forms and Sheets, namely our original contribution. The adapted solution was successfully applied to a Calculus class with 100 students.*

1. Introduction

Automated Assessment (AA) has always been a challenging task, specially when performed totally online. Moreover, AA has been demanded increasingly but professors and teachers must endeavour to evaluate the sought after students' skills in a fair way. When carried out manually, the process of elaborating and correcting questions turns out to be lengthy, time consuming and often subjective (Kosh et al., 2019; Choi and Zhang, 2019).

In order to circumvent this difficulty, Pugh et al. (2016) presents an Automatic Item/question Generator (AIG) for multiple-choice exams in Medicine. In Gierl et al. (2012) the authors present another cognitive process in Medicine handled by AIG. For language learning there also exists a great demand on AIG (Arendasy et al., 2012; Kim, 2017; Mo and Troia, 2017). A study in Kosh et al. (2019) analyzes the cost-benefit of either using a database of questions to develop Information and Communications Technology (ICT) for AIG, or resorting to the elaboration of all questions manually. Namely, the paper seeks to answer the question: *how many items ought to be produced before AIG becomes more efficient than the traditional writing of items?* This was a study in the context of multiple-choice maths questions for primary and secondary schooling, and it suggests that AIG becomes more practical than the handwriting for the threshold of 173 questions in the same content subject. After having been implemented, AIG becomes more advantageous already for producing at least 88 questions.

In Choi and Zhang (2019) the authors present a tutorial of a commercial ICT for creating questions by means of an Automatic Adaptive Formative Assessment available

[*]Grant #2018/23561–1, São Paulo Research Foundation (FAPESP).

on the web, that even includes voice recognition. There are many ICTs for AA, and here we cite Moodle, an open source learning management system (moodle.org), which also offers AA by means of AIG with *parametrized questions* (or *calculated questions*[1]). However, on this platform only some parameters can store random values (numbers or formulas), which hampers and restricts generating many variations of the same question with their respective answer keys. This makes impossible to include sophisticated questions with symbolic-numeric computations such as the algebraic derivative of an equation, the indefinite integral of a function, solving complex sorting and optimization problems, etc.

Like most AIGs the one proposed here produces questions with some variations, then uses computational resources to draw some parameters, shuffle the questions, and group them according to scope and difficulty. One can also use resources like Item Response Theory (IRT) in order to improve the calibration of each question's level of difficulty (Aybek and Demirtasli, 2017).

The aforementioned references give some empirical evidence of the usefulness of online AA to improve students' skills and to circumvent plagiarism. Furthermore, the solution presented here provides a practical tool to generate individual exams to each student, even at different classes, and to distribute the PDF files of those exams by email. This is an important contribution because it is an efficient, open source solution which employs free services (Google Forms and Sheets) where the student fills out an online form with the answers. It is also important as a strategy to circumvent plagiarism because each student receives a distinct exam.

The main purpose of this work is to present and evaluate an improvement of MCTest consisting of a web platform that aggregates databases into a single databank devoted to Education Systems and focused on Student Assessment. On this platform online tests and exams can be generated and corrected automatically through Google Forms and Sheets.

2. Using adapted MCTest: materials and steps

As mentioned before, the online evaluation study introduced in this paper was developed through adjusting an open source system called MCTest (see vision.ufabc.edu.br for details), which generates and corrects student activities automatically.

2.1. MCTest's development

We took version 5 of MCTest, the most recent one, developed on the web platform Django 2.2 (djangoproject.com) with Python 3.6, both installed on a web server. MCTest is free (github.com/fzampirolli/mctest), open source and offers an MYSQL format to Question Banks (QB) of Education System focused on student evaluations, specially with questions applied to teaching activities, in which exams are both generated and corrected automatically. According to MCTest's copyright, improvements achieved by the GitHub community must be included on this platform.

2.2. MCTest software

MCTest software enables the efficient preparation of dissertation or multiple-choice questions that can be parametrically modified, which produces many variations of these ques-

[1] docs.moodle.org/en/Calculated_question_type.

tions. This is accomplished by means of various libraries of Python programming language, with which MCTest computes answers automatically. In `vision.ufabc.edu.br` the reader will find some examples from already published works, as Zampirolli et al. (2019), which presents the parametric questions in MCTest.

Navigating through MCTest requires first logging in `vision.ufabc.edu.br` with permission, and there we are able to create or keep entities like Institution, Course, Discipline, Topic, Class, Exam and Question. For the time being any MCTest user is called Professor, who can also coordinate a Discipline and has permission to include Professor(s) and Topic(s) therein. Students are included in Classes through a CSV file that contains Id, Name and Email. Questions can be either uploaded in TXT files of MCTest's earlier versions, or included in the own forms of the system.

One of the sub-pages of MCTest is devoted to the entity Exam, on which the user can configure the test to be sent to students through email as an activity to prepare them for the exam in the classroom. Before the pandemic corrections required to digitize all the students' exams into a single PDF, which was then uploaded to MCTest. Afterwards, MCTest emailed a CSV file to the professor containing the correction of all multiple-choice questions in that PDF.

Compared with other automatic evaluation platforms, MCTest has the advantage that it enables parametric questions, which the user creates with both LaTeX and Python, specially by including some of its various libraries, for example SymPy (`sympy.org`).

Figure 1 shows an example whose outcome will be an exam in which only the equation of y is given and one has to compute its derivative y'. Characters between "[[code:" and "]]" are the parametric variables. Here $a1$ to $a5$ are the alternatives (always put the correct one in the field $a1$). Figure 2 shows an output of MCTest for this question.

2.3. Challenge

MCTest was originally projected for evaluations in a classroom with hardcopy tests. However, the world is undergoing a pandemic in 2020 that prevents us from meeting up. Therefore, we propose an adapted solution for MCTest to generate exams and email each of them in PDF to the respective student. Correction is performed in a spreadsheet (we resorted to Google Forms for the students to fill out with their answers, which are then forwarded to the professor's Google Sheets).

2.4. Contribution to the state of the art in ICT

In order to enable correction of exams generated by MCTest and emailed to students, we had to create a Google Form linked to Google Sheets where one can make changes as explained in this subsection. For the correction of dissertation questions we had to change the MCTest code, as explained at the end of this section.

2.4.1. Creating a Google Form

In order to correct an online exam we created a Google Form at `forms.gle/PgFH7w9mo9zWteTX8`. The form can be configured in such a way that students must access it with a valid email address, and also submit the filled-out form just once.

Figure 1. Interface to create/update a question (part I). The button Create-PDF shows what the question will be like in PDF. The button Save-Json stores it in JSON format. Choose Topic, Short Description, Group and Description are for the text that will be produced on the first line followed by the Python code. The other fields are self-explanatory.

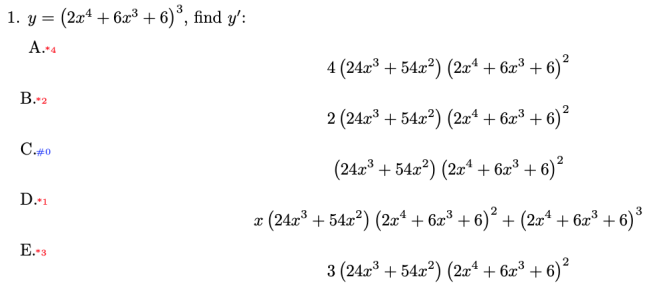


Figure 2. Interface to create/update a question (part II). Output of MCTest **for that question by clicking on Create-PDF. In this example alternative C came out as the correct one. Numbers in blue and red show the order after shuffling alternatives.**

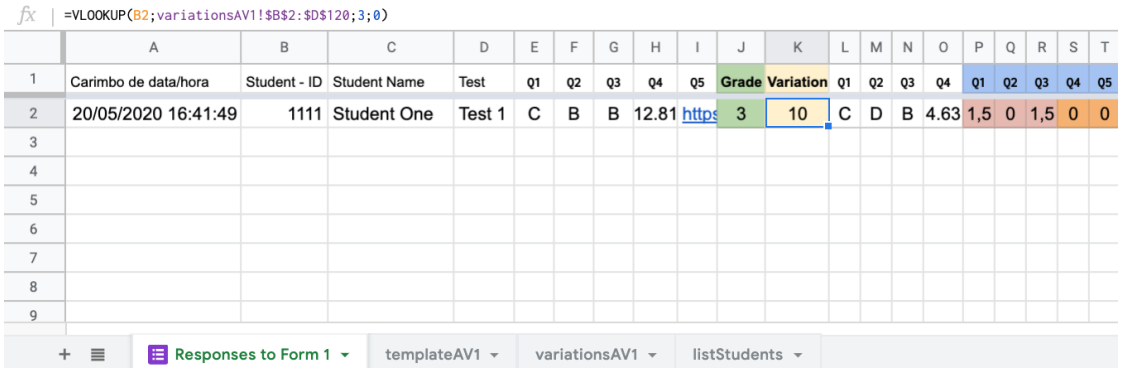
2.4.2. Configuring Google Sheets

We had to create Google Sheets for MCTest to carry out automatic correction of exams with parametric questions. See `url.gratis/TI4Si` and Figures 3, 4 and 5. Google

enables us to link a Form to a spreadsheet, hence the student submits their test in Google Form, which will update the spreadsheet automatically. See Figure 3, columns from A to I, where we take an exam with three multiple-choice questions, a written response question, and a final one for the student to send a file that, for instance, depicts a scanned handwritten answer, their id and signature.

We had to adapt the spreadsheet for it to correct the students' exams automatically. In Figure 3 column J computes the student's mark, which considers their emailed test issue displayed in column K. The mark is computed via the student's id in column B, together with the corresponding test issue stored in the tab `variationsAV1` (see Figure 5). Moreover, the final mark is obtained by accessing the tab `templatesAV1` in Figure 4, which contains the answer keys for each corresponding issue (notice that the value of Question 4 depends on the issue). In this example we have ten distinct issues of the same test.

Columns B to E in Figure 4 are reproduced in columns L to O in Figure 3, respectively, which is the answer key of issue 10. In this figure columns P, Q and R display the student's score for the three multiple-choice questions, which weigh 1.5 each. Another score is displayed in column S, namely of the forth question, whose answer key appears in column O. The score would be 2 if both numbers coincided. Finally, one has to check the student's photo in column I and attribute at most 3.5 to it in column T, since this Q5 is the scanned handwritten solution of the fifth parametric question.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	Carimbo de data/hora	Student - ID	Student Name	Test	Q1	Q2	Q3	Q4	Q5	Grade	Variation	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q5
2	20/05/2020 16:41:49	1111	Student One	Test 1	C	B	B	12.81	https	3	10	C	D	B	4.63	1,5	0	1,5	0	0
3																				
4																				
5																				
6																				
7																				
8																				
9																				

Figure 3. Sheet (part I). This one receives the students' data and answers, columns A to I.

	A	B	C	D	E	F
1	variation	Q1	Q2	Q3	Q4	Q5
2	1	C	E	D	3.47	
3	2	B	D	B	3.99	
4	3	A	B	C	4.07	
5	4	D	E	C	4.11	
6	5	E	B	A	4.29	
7	6	E	A	C	3.94	
8	7	B	B	D	3.56	
9	8	B	B	D	4.16	
10	9	B	E	B	3.73	
11	10	C	D	B	4.63	

Figure 4. Sheet (part II). Tab `templateAV1` from Figure 3 with issue numbers and answer keys, generated by MCTest **and emailed to the professor in a CSV file as soon as** MCTest **renders a PDF of the exam.**

	A	B	C	D
1	Room	ID	Name	Variation
2	Room1	1111	Student One	10
3	Room1	2222	Student Two	8

Figure 5. Sheet (part III). Tab `variationsAV1` from Figure 3 with the issue number sent to each student. This tab arises when the professor copies and pastes the CSV file sent by MCTest.

2.4.3. Adaptations of MCTest to dissertation questions

MCTest's code had to be extended for enabling the professor (or teacher) to receive two CSV files, as rendered in Figures 4 and 5, the one for the answer key of each test issue, and the other for the issue number (variation) sent to each student, respectively.

Moreover, we had to adapt the code for creating questions, as presented below, lest the professor want automatic correction (e.g. of Question 4 depicted in Figure 6).

```
A [[code:L0]] meter long wire should be cut into 2 parts (left
and right). With one of them to form a circle, with the other a
square. How should the wire be cut so that the sum of the areas
is minimal, considering that the left part is devoted to the
figure of least area? (Use $\pi=3$). \textbf{NOTE:} The answer
must be numeric with two decimals.

%%% Answer of a dissertation question to include in template
%%{ [[code:resp]] }%%  <<< use exactly this syntax
[[def:
import random
L0 = random.randrange(80, 110, 1) / 10
pi = 3
resp = "%.2f" % (L0*pi/(4+pi))
]]
```

In the description of this question we inserted the answer computed between "[[def:" and "]]", namely in the variable "resp". This variable was placed between "%%{" and "}%%". MCTest takes this parametric answer defined by "[[code:resp]]" and inserts its alternative into the corresponding cell visible in Figure 4, which actually comes from a CSV file. An MCTest output of this question is depicted in Figure 6.

1. A 8.1 meter long wire should be cut into 2 parts (left and right). With one of them to form a circle, with the other a square. How should the wire be cut so that the sum of the areas is minimal, considering that the left part is devoted to the figure of least area? (Use $\pi = 3$). **NOTE:** The answer must be numeric with two decimals.

Figure 6. An MCTest output of the dissertation question (adapted from Stewart (2006)).

3. Experience report

We have applied MCTest to a class of the discipline of Calculus that was attending the course Functions of a Unique Variable (FUV) at the Federal University of ABC. We shall present the context in which experiments with MCTest were applied, the obtained results and finally some discussion about our study.

3.1. Context of the experiments

FUV belongs to the programme Bachelor in Science and Technology (BST), which takes three years with three trimesters each. There the first course is called Foundations in Maths, devoted to levelling the students for the upcoming courses, and FUV is given in the sequel for the next trimester. The first trimester of 2020 was the third of the freshmen, and we have got 903 students matriculated in FUV (many attended FUV in the previous trimester but failed). By taking into account that BST admits circa 1,700 students every year, the failure rate in FUV is high (50.05% between 2013 and 2018). These 903 students were grouped in 10 classes. FUV is lectured four hours a week with two hours in different days. Each trimester consists of twelve weeks and in 2020 we had five in the classrooms before suspending lectures because of the pandemic. Afterwards the university took three weeks planning how to adapt courses to long distance. For example, in order not to affect students who have little access to the internet, every activity must allow 24 hours for them to finish. Since five out of twelve weeks were accomplished, the seven remaining weeks are to be completed as long distance courses. When we shall be back to the classrooms another three weeks will count for everyone to finish all exams.

3.2. Experiments

In the just described context we have performed experiments that will be presented now, and they refer to a class of FUV with 100 matriculated students. The professor provided video lectures for seven weeks, all devoted to distance learning activities. He gave two formative tests (Test1 and Test2) in order to help students be prepared to the evaluation exams (Exam1 and Exam2). Both test counted as a bonus of 10% on the final grade. They were all given in the same format: five parametric questions according to the aforementioned model (two of written response and three of multiple-choice). These were similar to the one in Figure 2, drawing the order of both the questions and their respective alternatives, including the parameters with various random values, as illustrated in Figure 1. The written response part consisted of Question 4, in which the student had to type a decimal number (see Subsection 2.4.3), and Question 5 for the student to solve by handwriting on a sheet of paper, signing, scanning and sending together with an image of their id.

Figure 7 shows the first page of an exam in PDF. The student had to finish it by filling out a specific form whose internet link is indicated in the exam instructions, and this link will render the web pages.

Test1 happened on a Monday for the students to become familiar with this new procedure, and they had 24 hours to hand in answers. Exam1 was emailed to the students on the Thursday of the same week, also to be solved in 24 hours.

A total of 62 students solved Test1, which on the one hand altogether had only 14 alternatives marked incorrectly (14/186=0.07). On the other hand Question 4 had only 6 correct answers, where the students could write on the form. In the case of Question 5, for the student to send the solution with a photo, they all did it and nobody claimed to have had technical problems during the submission.

Exam1 was really worthier for the final mark, and there the questions were pretty more elaborate, so that the student had to interpret the statements, similarly to the example in Figure 6. A total of 74 exams were sent, with 31 (31/222=0.13) wrong answers for the multiple-choice questions. But 42 students solved Question 4 correctly, and here they had

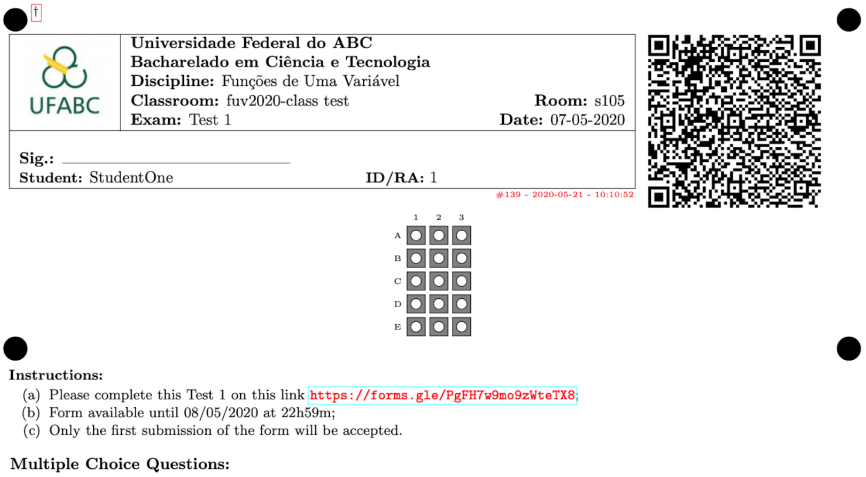


Figure 7. The beginning of a first page of an exam in PDF.

to write a number with two decimals. Again they did not have any technical problem with Question 5.

A similar performance was attained in Test2 and Exam2, but the great difference of the students' performance resided between the multiple-choice and the written response parts, as indicated in Table 1. They achieved high scores in the former but just passable to median scores in the latter. We suspect the reason lies in the fact already mentioned in Subsection 3.1, namely the students had 24 hours to finish any activity. Since the professor was unable to control plagiarism, even with many variations of questions some students could have resorted to applets like Photomath (`photomath.net`) to obtain the right answers. Such a resource does not have the same effect for highly elaborated questions like the example in Figure 6, because it could only be solved after a careful interpretation, something that was mandatory for the written response part.

In order to keep students working the professor gave Exam3 that did use variations and Google Form. But this time it consisted only of written response questions, in which the student had to submit a scanned image of their handwritten answer, their id and signature.

Table 1 summarizes the students' performances in the three exams. The 2nd column shows how many of the 100 students scored at least 75% of either Q4 or Q5. Indeed, comparisons between written response and multiple-choice parts would be less meaningful if we considered the whole class. We recall that in Exam1 and Exam2 questions 1, 2 and 3 were parametric and of multiple-choice, their order was shuffled together with the respective alternatives, and each question was worth 1.5. Exam3 had only written response questions, which were also parametric.

3.3. Discussions

Plagiarism is a serious and recurrent problem (Seitenfus et al., 2019), specially in the case of online activities. Individualized tests diminish this problem, yet some extra precautionary measures are necessary, like watching students by webcam while they do the activity, and restricting the time of the exam. As an example, one could set two hours for a long test and also configure the form to go only forward. Moreover, if internet connections

Table 1. Students' performance in Exams 1, 2 and 3. Notice the high average score in the multiple-choice part compared with the written response part.

Exam	Students	Multiple-Choice	Q4	Q5	Average	STDEV
1	74	86%	56.7%	59.5%	7.5	2.12
2	68	91.6%	39.7%	66.2%	7.7	2.01
3	65				5.1	1.89

were excellent and all students had a computer with webcam, then one could open a virtual room to supervise the exams. But most of the students have smartphone, and some can exchange messages without being noticed.

However, we are all undergoing a state of exception, and therefore some tools ought to be applied to enable learning activities. Therefore, our study represents an important resource to the students.

Less than 5% of the students reported a problem with the PDF attached to their emails. In some configurations of Microsoft Outlook the attachment appears as "`ATT00001.bin`". After the first student's contact that reported this problem, they all got a message in order to replace the extension "`.bin`" with "`.pdf`".

By considering the experiments in Subsection 3.2 and the restrictions of an asynchronous evaluation, we observe that online questions ought to be elaborated very well, either of multiple-choice or written response type. This will hinder finding the right solution by means of an applet, as supposedly happened in the experiments. Of course, a fair evaluation will require the professor to correct scanned computations manually, but our study is a valuable tool to keep students active by means of multiple-choice mock-exams. In this case the professor can give many of them, because MCTest already has QB for many courses and performs automatic correction, so that the final results will be stored in the professor's spreadsheet. Participants could get a small bonus in order to stimulate solving the mock-exams.

4. Conclusions and future works

In Subsection 2.2 we presented a totally online study of automatic generation of exams individualized by parametric questions produced with LaTeX and Python, all written in the free system MCTest. After the professor (or teacher) creates an Exam, they click on Create-PDF. Lest the user approve the PDF, with a yes-button they can enable its automatic sending through email to all students in the Classes selected on the screen "Exam". Only the user gets an email with two CSV files, the one containing the answer keys of each test issue, and the other the drawn numbers of each test issue sent to the respective student.

When the student opens their PDF, in the header they will find a URL to a Google Form. When the student sends their answers through the form, these automatically go to the professor's Google Sheets. The automatic correction happens with very little intervention in this sheet. This study was applied to a Calculus class with 100 matriculated students. By considering the context in which the exams were given, namely students had 24 hours to hand in their forms, we consider that the study is satisfactory.

By taking into account that MCTest can generate infinitely many issues of each

question, as exemplified in Subsection 2.2, then it is better than many other AAs because it also computes the right answer automatically as seen in Subsection 2.4. By considering the cost-benefit, MCTest is highly advantageous. Moreover, MCTest is not just another AIG because it includes a complex web environment for AA in educational institutions.

Among the improvements that can be implemented, we call attention specially to security against plagiarism. MCTest can be endowed with encrypted facial recognition in the QRCode of each exam. Moreover, MCTest can gain new functionalities for a student user like question timer, automatic feedback of questions and mark computation, all these independently of Google Form and Sheets. By considering a scenario in which thousands of students are sitting an exam simultaneously, it would also be important for MCTest to automatically calibrate each question's level of difficulty through an Item Response Theory (IRT) (Aybek and Demirtasli, 2017). In this sense, MCTest already emails some statistics of the automatic correction of the digitized exams to the professor, and for that it uses IRT.

References

Arendasy, M. E., Sommer, M., and Mayr, F. (2012). Using automatic item generation to simultaneously construct german and english versions of a word fluency test. *Journal of Cross-Cultural Psychology*, 43(3):464–479.

Aybek, E. C. and Demirtasli, R. N. (2017). Computerized adaptive test (cat) applications and item response theory models for polytomous items. *International Journal of Research in Education and Science*, 3(2):475–487.

Choi, J. and Zhang, X. (2019). Computerized item modeling practices using computer adaptive formative assessment automatic item generation system: A tutorial. *The Quantitative Methods for Psychology*, 15:214–225.

Gierl, M. J., Lai, H., and Turner, S. R. (2012). Using automatic item generation to create multiple-choice test items. *Medical education*, 46(8):757–765.

Kim, E.-Y. J. (2017). The toefl ibt writing: Korean students' perceptions of the toefl ibt writing test. *Assessing Writing*, 33:1–11.

Kosh, A. E., Simpson, M. A., Bickel, L., Kellogg, M., and Sanford-Moore, E. (2019). A cost–benefit analysis of automatic item generation. *Educational Measurement: Issues and Practice*, 38(1):48–53.

Mo, Y. and Troia, G. A. (2017). Similarities and differences in constructs represented by us states' middle school writing tests and the 2007 national assessment of educational progress writing assessment. *Assessing Writing*, 33:48–67.

Pugh, D., De Champlain, A., Gierl, M., Lai, H., and Touchie, C. (2016). Using cognitive models to develop quality multiple-choice questions. *Medical teacher*, 38(8):838–843.

Seitenfus, D., dos Santos, B. V., Manica, E., and Pertile, S. (2019). Percepção de plágio acadêmico entre estudantes e professores de cursos de graduação e pós-graduação na modalidade a distância. *RENOTE-Revista Novas Tecnologias na Educação*, 17(1):103–112.

Stewart, J. (2006). Cálculo, vol. 1, 5ª edição. *Editora Thompson*.

Zampirolli, F., Teubl, F., and Batista, V. (2019). Online generator and corrector of parametric questions in hard copy useful for the elaboration of thousands of individualized exams. In *CSEDU (1)*, pages 352–359.