Gradiance On-Line Accelerated Learning Guide for Authors

Jeffrey D. Ullman
Gradiance Corp.

Abstract

Gradiance On-Line Accelerated Learning (GOAL) is a system for creating and automatically grading homeworks, programming laboratories, and tests. Through the concept of "root questions," Gradiance encourages students to solve complete problems, even though the homework appears to be in a multiple-choice format. This document contains advice about how to design effective root questions. The mechanics of entering questions into the Gradiance system are found in the companion "Instructor's Guide."

Contents

1 What is a Root Question?

A root question looks to the student like a multiple-choice question, but there is much more behind the scenes. As discussed in the instructor's guide (`www.gradiance.com/pub/inst-guide.pdf`), a root question asks the student to solve a conventional problem, but then tests their knowledge by asking them to identify a piece of the correct solution from among four random choices. We ask students to solve a group of problems together and invite them to keep trying the work until they get a perfect score. When a student is able to answer correctly a multiple-choice question about every one of the problems in a group of 5 or so, we can be reasonably confident that the student really has mastered these problems.

To further turn the homework into a learning, rather than testing, process root questions generally have "choice explanations" (CE's) associated with each incorrect choice. The CE is shown to the student whenever they make that choice and offers some advice about how to solve the problem. We discuss effective CE's in Section 2.2.

1.1 Components of a Root Question

A root question requires the following components:

1. The *stem* — the statement of the problem.

2. The solution. This explanation is shown to the student after the homework due-date has passed and corresponds to a solution sheet that might be distributed after a conventional homework is finished.

3. Some number of correct choices. This number may vary, but we recommend 3-5. Remember that the Gradiance system will choose one of these at random, along with three random incorrect choices, and present the four choices to the student in random order.

4. Some number of incorrect choices. There should be approximately three times as many incorrect choices as correct choices, so it is not too easy for a student, trying the same homework several times, to notice the different frequency of correct and incorrect answers.[1] In practice, it is safe to deviate significantly from the 1:3 ratio.

5. Choice explanations. You have the option of adding choice explanations for any choice, but it is important to add CE's for at least the incorrect choices. When a question is phrased in the negative ("which of the following is *not* a gizmo?") then it may make sense to provide CE's for the correct choices, explaining why they are not gizmos.

2 Root-Question Design

The most important point to remember when designing root questions is that you are really creating a very conventional "analytic" question, where the student is asked to solve some problem in Math, Science, or Engineering. Your stem asks the students to solve the problem. However, instead of having them write down their solution and submitting it on the due date to be graded and

[1] Interestingly, even if there are exactly three times as many incorrect answers as correct, a student who keeps track of co-occurrence of answers over many iterations of the homework can discover the correct answers. However, this possibility is too remote to worry about. Gradiance limits the frequency with which a student can open a homework to avoid gathering of meaningful statistics by the student.

returned a week after that, you will ask them to answer a random multiple-choice question about their solution and give them immediate feedback. Thus, you need to identify components of the correct solution that the student can identify if they have solved the problem correctly.

We are going to work through a typical example, using integral calculus as the domain. Our goal is to make sure that students understand the rule for integrating polynomials, i.e., $\int x^n = x^{n+1}/(n+1)$. A conventional problem would be something like:

> What is the indefinite integral of $20x^4 + 12x^3 + 30x^2$?

To turn this question into a root question, we need to observe that there are three terms to this polynomial, so there are three natural components that lead to three correct choices. If we wanted more choices, we could add more terms to the polynomial. Thus, we can phrase the root question as:

> Compute the indefinite integral of $20x^4 + 12x^3 + 30x^2$. Then, identify one of the terms in the integral from the list below.

2.1 Choices

There are three correct choices in this example: $4x^5$, $3x^4$, and $10x^3$. We should now develop approximately nine incorrect choices. One approach is to think of the common mistakes that a student might make, and make those be incorrect choices, coupled with a CE that addresses the assumed mistake. In our example, one common mistake is to forget to divide by $n+1$. This mistake leads to incorrect choices $20x^5$, $12x^4$, and $30x^3$. Another possible mistake is to divide by n instead of $n+1$. Thus, we might choose $5x^5$, $4x^4$, and $15x^3$ as incorrect choices. We could proceed by theorizing about other possible errors, or just add some random, plausible looking incorrect choices such as $3x^3$, $60x$, $80x^5$, and $6x^3$. We now have three correct and 10 incorrect choices, a reasonable combination.

2.2 Choice Explanations

In general, a CE should do at least one of the following:

1. *Explain why the choice is incorrect.* Ideally, the reasoning used to explain the error also hints at the general methodology for solving the problem, by taking the student through the key steps.

2. *Outline a solution to the problem.* In this case, the CE for each choice is the same. A variant is to break the outline into a series of hints. Attach one hint to each incorrect choice. If the student gets the same question wrong many times, he or she will accumulate a progressively larger collection of hints.

If the question is designed for a particular textbook, we could even add a citation of a place to read in the text.

2.3 Example: The Integration Problem

For the incorrect choices that were designed to detect specific mistakes, the natural CE will point out that mistake. For example, the CE associated with the incorrect choices $20x^5$, $12x^4$, and $30x^3$ might say:

> The correct rule for integrating polynomials requires that we divide the term by a constant. Do you remember how that constant is determined?

For the incorrect choices $5x^5$, $4x^4$, and $15x^3$ we might say something similar:

> The correct rule for integrating polynomials requires that we divide the term by a constant. However, you may have chosen the wrong constant.

These CE's simply try to jog the memory of students who may understand the idea but are careless. Since students may be lost, we might choose to attach to the remaining four incorrect choices a more explicit piece of advice: the general solution to the problem. For example:

> In order to integrate a polynomial, we integrate each term and sum the results. The rule for integrating a term is $\int ax^n = ax^{n+1}/(n+1)$.

2.4 Another Example: Understanding a Finite-State Machine

Let us consider a problem that might appear in a logic-design course in EE or an automata course in CS: understanding what a simple finite-state machine (FSM) does. The FSM has two states, A and B; A is both the start and accepting state. It has the following transition table:

	0	1
A	A	B
B	B	A

If you are familiar with state machines, you can notice that the machine stays in the same state on 0 inputs (i.e., it ignores 0's) and switches state whenever it sees a 1. It is thus counting the number of 1's it sees, and accepting whenever that number is even. A standard phrasing of the question would be:

> Give a simple English statement describing what the FSM above does.

To make it a root question, we need to dissect the answer to the above question into atoms; in this case the "atoms" are the individual input strings that the FSM might or might not accept. That is:

> Examine the FSM above, and determine what it does. Demonstrate your understanding by identifying, from the list below, the string that this FSM accepts.

Then, we can choose as correct answers any string of 0's and 1's with an even number of 1's, e.g., 010011010101. We can also choose as incorrect answers any strings with an odd number of 1's, e.g. 01001101001.

If the student has figured out what the FSM does, then they can easily scan any four choices to find the one with an even number of 1's. If they have not figured it out, but have patience, they can simulate the FSM on each of the four strings. But that becomes painful, since they may have to work the problem several times (because they need to repeat *other* questions in the same homework if they have made any mistakes), so they have an incentive to solve the problem.

We also encourage the student to solve the complete problem by our choice of CE's. A basic CE would show why a choice is wrong by simulating the FSM. For instance, if 010 were a choice, we might say:

On input 010, the FSM makes the following transitions: $A -0\rightarrow A -1\rightarrow B -0\rightarrow B$. As a result, it winds up in state B and does not accept.

In addition to reminding the student of what acceptance means, and how a FSM processes input, it convinces the student that the answer is wrong.

However, we can use, in addition to or in place of this kind of argument, a hint about the overall working of the FSM . These hints can appear with all incorrect choices, or perhaps scattered around the choices, so the more times a student gets the question wrong, the more hints they are likely to have at their disposal. Examples of hints are:

> Notice that the FSM changes state whenever the input is 1 and stays in the same state whenever the input is 0.
>
> Consider what the FSM does on the following inputs: 0000, 1001, 0110, and 1111. Do you notice a pattern?
>
> Hint: the FSM is, in a sense, counting 1's.

3 Using Parameters

Gradiance allows you to use text-valued parameters in questions. Having variants of the question with different values of the parameters can sometimes substitute for the variety of choices that occur in a conventional root question. We represent parameters by `%%1`, `%%2`, and so on; any integer may be used. You may examine the instructor guide for the mechanics of entering the parameter-value sets, but here we shall give some examples of their use.

3.1 Example: Parameterized Question on Integration

We can phrase the question from Section 2 more normally as:

> What is the integral of $\%\%1x^{\%\%2}$?

There will be one correct choice, which we could write as $\%\%3x^{\%\%4}$.[2]

One incorrect choice could be $\%\%1x^{\%\%4}$, with the choice explanation:

> Do not forget that when you integrate a polynomial, you must divide the coefficient by a certain constant.

Another choice could be $\%\%5x^{\%\%4}$, with the intent that %%5 is %%1/%%2, an "off-by-one" error, and a third choice might be $\%\%3x^{\%\%2}$, catching a student who forgets to raise the exponent.

Note that in this situation, we do not need to have more than one correct and three incorrect choices. However, we are allowed to create any number of correct and incorrect choices, even though

[2]In a future release of the Gradiance system, you will have the ability to specify arithmetic computation on parameters. Then, you will not need %%4, since it could be expressed as ##%%2 + 1##. Likewise, %%3 could be expressed as ##%%1/(%%2 + 1)##, but you may prefer not to do so. The reason a separate parameter %%3 is useful is that, depending on whether the quotient of %%1 and %%2 + 1 is an integer, a finite decimal, or an infinite repeating decimal, you may prefer different forms for %%3, e.g., 2.5 or 1/3. In fact, the case of the integral of x^2 (or any term with coefficient 1) is sufficiently unusual that it almost requires us to use a third parameter for the result. Notice that for x^2, %%1 must be the empty string, not 1; %%2 is 2, of course. However, the empty string divided by 3 is meaningless.

parameters are used. If we have extra choices, then each time the student is shown the question, a random parameter-value set is chosen; then a correct and three incorrect choices are picked at random, and finally the choices are ordered randomly.

The question is completed by providing a stock of parameter-value sets. In this example, as in many questions, there are relationships among the parameters; specifically:

- %%3 = %%1/(%%2 + 1).
- %%4 = %%2 + 1.
- %%5 = %%1/%%2.

Some possible entries in the parameter table are:

%%1	%%2	%%3	%%4	%%5
2	3	0.5	4	(2/3)
3	6	(3/7)	7	(1/2)
	4	0.2	5	0.25

4 Problematic Areas for Designing Root Questions

We shall now take up two techniques for using root questions in arenas to which they seem hard to adapt:

1. Algebraic rules and transformations.
2. Proofs.

4.1 Algebra

In several interesting domains we would like to use questions that involve transforming expressions or determining whether a proposed equality is in fact an equivalence. For example, to teach matrix algebra, we might ask students to determine the truth or falsehood of equations such as $(\mathbf{A}^T)^T = \mathbf{A}$ (true) or $\mathbf{AB} = \mathbf{BA}$ (false). Our first attempt at a root question would be to ask:

> Which of the following equations is true?

or perhaps

> Which of the following equations is false?

Either way, the student will be presented with new equations to work out each time they open the homework; that situation is hard on students, as discussed in Section 2.4.

A better approach is to limit the scope of the problem by giving the student a fixed set of expressions to consider. Ask them to discover all equivalences, and identify from a list of four the pair of equivalent expressions. Here is a simple example:

> Let $\mathbf{A}$ and $\mathbf{B}$ be $n \times n$ square matrices and let $\mathbf{I}$ be the $n \times n$ identity matrix. Consider the following matrix expressions: (a) $(\mathbf{AB})^T$ (b) $\mathbf{A}(\mathbf{I} + \mathbf{B}^T)$ (c) $\mathbf{A}^T\mathbf{B}^T$ (d) $\mathbf{A} + \mathbf{AB}^T$ (e) $\mathbf{B}^T\mathbf{A}^T$ (f) $\mathbf{AB}^T + \mathbf{IA}$. Determine which pairs of expressions are equivalent, and select from the list below the equivalent pair.

You will notice that $b = d = f$ and $a = e$, but there are no other equivalences. Thus, there are four correct answers and six incorrect answers — not an ideal ratio, but good enough. This question is a good one to put CE's on all choices, correct and incorrect. Correct choices can have a proof of equivalence, and incorrect choices can offer a counterexample.

4.2 Proofs

Proofs are hard to turn into any sort of mechanically graded question. One technique with which we've had some success is to offer students an outline of a proof and ask them to identify the reason for each step. The theory is that if they can supply reasons, then they probably understand the proof, even if they couldn't come up with it themselves. For instance, we can ask students to distinguish among uses of the inductive hypothesis, and uses of various definitions of concepts involved in the proof. We shall illustrate the idea with the set-up actually used for several questions involving context free grammars.

Let G be the grammar:

$$S \rightarrow SS \mid (S) \mid \epsilon$$

$L(G)$ is the language BP of all strings of balanced parentheses, that is, those strings that could appear in a well-formed arithmetic expression. We want to prove that $L(G) = BP$, which requires two inductive proofs:

1. If w is in $L(G)$, then w is in BP.
2. If w is in BP, then w is in $L(G)$.

We shall here prove only (1). You will see below a sequence of steps in the proof, each with a reason left out. These reasons belong to one of three classes:

A) Use of the inductive hypothesis.

B) Reasoning about properties of grammars, e.g., that every derivation has at least one step.

C) Reasoning about properties of strings, e.g., that every string is longer than any of its proper substrings.

The proof is an induction on the number of steps in the derivation of w. You should decide on the reason for each step in the proof below, and then identify from the available choices a correct pair consisting of a step and a kind of reason (A, B, or C).

Basis: One step.

(1) The only 1-step derivation of a terminal string is $S \Rightarrow \epsilon$ because ________

(2) ϵ is in BP because ________

Induction: An n-step derivation for some $n > 1$.

(3) The derivation $S \Rightarrow^n w$ is either of the form
(a) $S \Rightarrow SS \Rightarrow^{n-1} w$ or of the form
(b) $S \Rightarrow (S) \Rightarrow^{n-1} w$
because ________

Case (a):

(4) $w = xy$, for some strings x and y such that $S \Rightarrow^p x$ and $S \Rightarrow^q y$, where $p < n$ and $q < n$ because ________

(5) x is in BP because ________

(6) y is in BP because ________

(7) w is in BP because ________

Case (b):

(8) $w = (z)$ for some string z such that $S \Rightarrow^{n-1} z$ because ________

(9) z is in BP because ________

(10) w is in BP because ________

One form of question is to simply ask for those steps that use the inductive hypothesis. The correct answers are (5), (6), and (9); the other seven choices are incorrect. Another form is to ask for correct pairs of a step and a reason (A, B, or C). Then, choices such as $(5, A)$, $(3, B)$, or $(2, C)$ are correct; choices such as $(5, B)$ or $(5, C)$ are incorrect. There are 10 correct and 20 incorrect answers; we don't have to use all of them, of course.