The WebAssign Answer Evaluator

WebAssign is an online homework system that delivers instantaneous grading and feedback. Students using WebAssign enter their answers to exercises using traditional mathematical notation. This means that students communicate with the WebAssign system by using the language of mathematics.

The language of mathematics is a lot like the English language. It is always possible for a speaker of good English to clearly and unambiguously say what he or she means. However, there is nothing stopping a novice English speaker from uttering ambiguous sentences like

"I saw a man with a telescope."

In this sentence, it is not clear whether the telescope belongs to the person observing or the person being observed. Similarly, statements written in the language of mathematics can be ambiguous. Consider the relatively simple statement “\( \sin x \cdot y \)”. Does this statement mean “\((\sin(x)) \cdot y\)” or “\(\sin(x \cdot y)\)”? 

This document is an introduction to the answer evaluation and grading techniques that affect students' responses to exercises in WebAssign. The processing is complex, but the end result is simple: WebAssign provides students an intuitive experience wherein the focus remains on learning the course material, not learning a homework system.
The WebAssign Answer Evaluation and Grading System

The Challenge
One of the main applications of an online grading system like WebAssign is to grade so-called “free response” questions. Take, for example, the following exercise and corresponding answer key.

Exercise: Find the indefinite integral.
\[ \int \frac{1}{x} \, dx \]

Answer Key: \(-\ln(x) + C\)

In order to deliver and grade this question with an online system, the software must be able to accept many different variations on the answer key. For instance, if a student wrote his or her response as
\[ \ln \left( \frac{1}{x} \right) + C \]
he or she should still be awarded credit. After all, the student has only applied the laws of logarithms to rewrite the expression in a mathematically valid way. Notice also that the student has not used parentheses to explicitly delimit the argument to the natural log function. If the student wanted to use parentheses, such as
\[ \ln \left( \frac{1}{x} \right) + C \]
his or her answer should still be accepted. In general, there will be infinitely many ways to write equivalent forms of the required expression, and the homework system cannot predict what form the student’s answer will take.

One traditional workaround for this problem is to force students to enter their answers in a specific form. Unfortunately, this introduces many new problems. Foremost, restricting responses to a specific form destroys the “free response” aspect of the question that is so important pedagogically. Also, a
homework system that accepts only specific forms of a response may cause usability issues. Further, this restriction may mislead students into believing their equivalent response is mathematically incorrect.

Taking the idea of equivalent and correct responses even further, we wonder if responses like

$$\frac{1}{x} + 2c$$

should be accepted as well. After all, this answer is no less general than the one shown in the answer key, since $c$ is assumed to be any arbitrary real number. If answers like these are to be accepted, then

$$\frac{1}{x} + \ln c$$

should also be accepted, since any real number can be written as $\ln c$ for some number $c$. Therefore, even answers like

$$\ln \frac{c}{x}$$

should be accepted for the original answer key of

$$-\ln(x) + c$$

In situations like this, the student’s response should be considered correct if and only if it satisfies some specific mathematical conditions. The original answer key is merely one example of many correct answers. Questions requiring this sort of grading occur frequently in all areas of mathematics instruction.

A similar problem arises when numerical approximations are allowed. Take, for example, the following question and answer.

**Exercise:** Use a graphing utility to approximate a sinusoidal function with period 4 which has a peak at $(1, 3.7)$, and a valley at $(3, 1.7)$.

Figure 1
Answer Key: \( \sin\left(\frac{x}{2}\right) + 2.7 \)

Students who use a calculator to approach this problem may obtain answers like

\[ \sin 3.14x/2 + 2.69 \]

or even

\[ 2.7 + \sin 1.57x \]

Exercises involving long numerical calculations are common, especially in academic disciplines such as physics. The ability to accept expressions that are approximately right is crucial for an online homework system.

The challenge here is to provide a general framework for effectively delivering and automatically grading these types of “open-ended” questions.

**Strategy and Implementation**

After considering these and other real-world examples, the first goal of WebAssign was to accept student input and use the computer to process its mathematical meaning. The input can then be algorithmically tested for certain mathematical criteria. Those test results then allow WebAssign to output immediate assessment, grades, and feedback.

The critical step of this strategy was building a robust evaluator for traditional mathematical notation for WebAssign. After studying the notation styles used in various mathematics textbooks and publications, it became apparent that the language of mathematics is fundamentally different from a computer programming language. A programming language is purposefully designed and unlikely to change. In contrast, the language of mathematics has continuously evolved over the course of human history. As a
result, mathematical notation is extremely flexible and expressive; however, with that flexibility and power comes the possibility of ambiguity.

Ambiguity in this context is usually the result of different overlapping notational constructs applying to the same portion of an expression. For example, multiplication between two terms is commonly notated by separating the two terms by a space, as in

\[ x \ y = 1 \]

However, spaces are also commonly employed as argument separators when using functions without parentheses. The following example clearly shows both uses of the space character.

\[ x \ y = \sin(2z) \]

In order to allow students to write notation in this way, it was necessary to permit the computer to interpret the space character in either way, whenever appropriate. Now, imagine a student enters the following expression.

\[ \sin(x) \ y \]

It is not clear whether the student meant to use the second space as a separator, meaning

\[ \sin(x) \ y \]

or as multiplication between \( x \) and \( y \), meaning

\[ \sin(x \ y) \]

Ambiguities like these have the potential to appear in many different mathematical constructions. As another example, consider an expression like

\[ x + 1,234 + y \]

This expression could mean either

“\( x + 1234 + y \)”

or

“a two-element list consisting of the element ‘\( x + 1 \)’ and the element ‘\( 234 + y \)’”
Since both of these interpretations could represent valid answers in the right context, it is difficult for a computer to decide which one is “better” in any sense.

As more and more examples like these came up, it became apparent that in order to allow all academically accepted mathematical notations, the possibility of ambiguous notation had to be allowed as well by WebAssign. In this way, the language of mathematics is a lot like the English language. It is always possible for a speaker of good English to clearly and unambiguously say what he or she means. However, there is nothing stopping a novice English speaker from uttering ambiguous sentences like

“I saw a man with a telescope.”

This ambiguous sentence is similar to the ambiguous expression “\( \sin x y \)”: it is not clear whether the telescope goes with “man” or with “saw”!

Even though the language of traditional math notation has the potential for ambiguity, we still need a computer to evaluate it in a robust fashion. In the interest of preserving students’ natural notation styles as much as possible, we made the decision to simply deal with the ambiguity programmatically rather than force the computer to choose one interpretation over another.

Additionally, the mathematical syntax itself can help eliminate spurious interpretations before they are formed. For example, the character “i” has multiple meanings including the imaginary unit \( \sqrt{-1} \) and the 3-dimensional unit vector \(<1,0,0>\). But when a student inputs \( i + 2 \), we know to ignore the vector interpretation because one cannot add a scalar to a vector. In this way many spurious interpretations are avoided while still considering all salient interpretations.

Implementing the methods to allow for ambiguous responses was a major accomplishment, but there is still the crucial step of actually grading the student input. For example, consider a question that requires the derivative of the response to exactly match the derivative of the answer key. In this scenario, the student response and the answer key can be passed to an auxiliary computer algebra system and differentiated symbolically.

**Conclusion**

The WebAssign answer evaluation and grading system offers solutions to many of the most difficult challenges associated with automatic homework assessment. Most importantly, the system helps WebAssign achieve two main goals: to help students learn and to help teachers teach.

From a student perspective, online homework systems like WebAssign are deemed effective only if the software is at least as easy to use as the traditional means of doing homework: pencil and paper. Well-designed input tools like mathPad are only the first step in crafting a strong learning environment – the
"behind the scenes" processing that occurs is critical as well. By permitting and processing all types of natural mathematical input, students answering homework exercises in WebAssign have an uncomplicated and intuitive experience. This design ensures students are not learning a homework system; instead, the focus properly remains on the real course material.

For instructors, the WebAssign design results in a trustworthy and pedagogically accurate assessment tool. The automatic grading attached to each WebAssign exercise is customized to be as strict or as lenient as the exercise's instructions and definitions demand, allowing the system to grade much more like an actual instructor than a computer.

The WebAssign answer evaluation and grading system ensures students receive immediate and accurate feedback. Overall, WebAssign provides the tools with which students may develop and demonstrate concept mastery. With these robust methods in place, WebAssign frees instructors to focus on teaching the course material.

**A Note Regarding Cengage Learning Enhanced WebAssign Titles**

Each question developed in WebAssign is carefully constructed to ensure that it best matches the textbook pedagogy. In cases where the textbook calls for an answer to be written in a specific format to match the textbook pedagogy, WebAssign will not accept an equivalent answer in a different format. Students will be provided with specific instructions in the question text on how to input answers in these cases.

For a complete list of available Enhanced WebAssign titles, please visit [http://webassign.net/cengage](http://webassign.net/cengage).