A Comparison of Equality in Computer Algebra and Correctness in Mathematical Pedagogy

Russell Bradford[1], James H. Davenport[1], and Christopher J. Sangwin[2]

[1] Department of Computer Science
University of Bath, Bath BA2 7AY, United Kingdom
{R,J.Bradford,J.H.Davenport}@bath.ac.uk
[2] Maths Stats & OR Network, School of Mathematics
Birmingham, B15 2TT, United Kingdom
C.J.Sangwin@bham.ac.uk

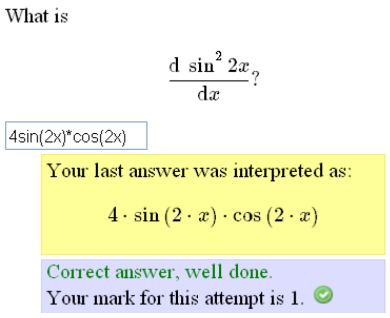
Abstract. How do we recognize when an answer is "right"? This is a question that has bedevilled the use of computer systems in mathematics (as opposed to arithmetic) ever since their introduction. A computer system can certainly say that some answers are definitely wrong, in the sense that they are provably not an answer to the question posed. However, an answer can be mathematically right without being pedagogically right. Here we explore the differences and show that, despite the apparent distinction, it is possible to make many of the differences amenable to formal treatment, by asking "under which congruence is the pupil's answer equal to the teacher's?".

1 Introduction

The purpose of this paper is to examine current computer aided assessment (CAA) practise from a theoretical computer science point of view. In particular, we envisage a student being asked a mathematical question in an online automated assessment system such as the following, "what is $\frac{\mathrm{d}\sin^2 2x}{\mathrm{d}x}$?". Such online assessments are becoming rather commonplace, and an example from the STACK system [27] is shown in Figure 1. Neverthless, the field is still at the 'craft' stage, and is bedevilled by the various meanings attached to the concept of "right answer". This paper aims to provide a more formal underpinning than has existed hitherto, which we hope will allow better communication, collaboration and understanding.

In this paper we are not concerned with the very real difficulties of mathematical input, see for example [29]. Note in Figure 1, for example, the student has not been diligent in making every multiplication explicit, although the feedback interpreting this one-dimensional string in traditional format has. We are instead interested in automatically establishing equality of two expressions. Henceforth, we assume we have valid parse trees which represent the student's answer.

While a computer algebra system can typically only say "right or wrong" (coded here as T or F), a teacher using CAA normally requires three outcomes. The first is a numerical mark (also called a score), which we will normalise to


What is

$$\frac{\mathrm{d}\ \sin^2 2x}{\mathrm{d}x}?$$

4sin(2x)*cos(2x)

Your last answer was interpreted as:

$$4 \cdot \sin(2 \cdot x) \cdot \cos(2 \cdot x)$$

Correct answer, well done.

Your mark for this attempt is 1.

Fig. 1. Typical computer aided assessment

be out of 1. The second outcome is feedback, which is text given to the student. Figure 1 shows these two outcomes. The last outcome is a *note* for the teacher. Typically in CAA questions are randomly generated and the feedback, if given, may contain manipulated expressions which depend on the random parameters, or student's answer. Hence, if the teacher wishes to generate statistics of the outcomes, the feedback and score are not helpful. Instead the note records the logical outcome, regardless of the exact question asked, or precise answer given. Some typical answers to the question in Figure 1 are given in Table 1.

No. 1 represents the correct answer, in the expected *form*. No. 2 evaluates to the correct answer, however in practice we would not allow restatements of the question like this as a valid answer. For this example, such behaviour could be by disabling the the 'derivative' key on the input palette or similar, but in other circumstances this is harder to prevent. When we consider the equality of two expressions, we do so here with the differentiation operator being a *noun* in the student's expression. In no. 3 and no. 4 it is, arguably, clear the student is operating along sensible lines, and so some partial credit has been awarded. Perhaps some feedback "don't forget that differentiating the square gives you a factor of two, *and* the factor of $2x$ gives you a factor of two" might be appropriate in no. 3. Regarding no. 5 as correct assumes we had configured our algebra system to use `trigreduce` or the equivalent, and this emphasises the importance of a relatively sophisticated algebra engine to mark even relatively simple exercises. In no. 7 the student has integrated by mistake, and this is a situation we can anticipate and for which we may provide helpful feedback, though this is outside the main scope of *this* paper.

There are many examples of CAA which evaluate student's answers in a sophisticated mathematical way. An early example using Maple was AiM [20,31]. While this system was, and remains, useful for assessing many questions, it cannot provide some kinds of detailed feedback, particularly at an elementary level. Other Computer Algebra Systems (CAS) have been used, e.g. the STACK system uses Maxima, see [27,28]. It is not necessary to use a *mainstream* CAS to process students' responses in a CAA system. For example, CALM (see [1]), Metric (see [24]) and Aplusix (see [9]) developed their own mathematical libraries. We argue these are "computer algebra" in its broadest sense. Fundamental to all these systems is the need to compare two mathematical expressions using computer algebra of some kind. This is the issue we examine in the remainder of this paper. We note that systems like WebWorK, [16], check for mathematical correctness by evaluating at a number of points. They are therefore essentially testing extensional equivalence and so do not fall within the scope of this paper, although in practice there is probably considerable scope for a hybrid approach.

Table 1. Typical human-marked answers

No.	Student's answer	C.A.	Score
1.	$4\sin 2x\cos 2x$	T	1
2.	$\frac{\mathrm{d}\sin^2 2x}{\mathrm{d}x}$	T	0
3.	$2\sin 2x\cos 2x$	F	0.7
4.	$2\times 2\sin 2x\cos 2x$	T	0.8
5.	$2\sin 4x$	T?	1
6.	$2\sin 2x\cos 2x + 2\sin 2x\cos 2x$	T	0.8
7.	$x/4 - \sin(4x)/8$	F	0

2 What Is a 'Right' Answer (Pedagogically)?

Ever since its introduction (probably in [25], see [6, (I), p. 165]), the sign '=' has had several meanings. There are at least six senses in which this synonym is currently used in traditional written notation (and even in computer algebra, equality has many meanings [12]):

(i) assignment of a value to a variable ($x = 1$);
(ii) to denote an equation yet to be solved ($x^2 + 1 = 0$);
(iii) definition of a function ($f(x) = x^2$);
(iv) as notation for λ-reduction, or combinatory reduction, as in "$\mathbf{K}MN = M$" [2, Corollary 2.1.26], and hence informally as in "what is 1+1 equal to?";
(v) as a "variant" of $\in$, as in $f(x) = O(x^2)$ [13, Section 8]; and
(vi) as a Boolean infix operator, returning either TRUE or FALSE.

It is not symmetric in uses (i), (iii) and (v), and not always in (iv).

It is the last sense we wish to examine in detail in this paper, since it is a crucial component in mathematical pedagogy, particularly in the assessment process. But here we are not concerned with = as an operator on statements of predicate logic, but in establishing the equality of mathematical expressions — yet another potential usage for this symbol.

Furthermore, we are concerned with getting, not merely “a correct” answer, but also “the right” answer. As a further example, let us assume a teacher has asked a student to

$$\text{expand out } (x+1)^2 \tag{1}$$

and the response they have from one student is $x^2+x+x+1$. This is “correct” in the sense that it is algebraically equivalent to $(x+1)^2$ and is in expanded form (actually two separate mathematical properties) but “incorrect” in the sense that the student has not *gathered like terms* by performing an addition $x+x$. We might say that the student has fallen at one of the two hurdles:

explicit task — do the expansion;
implicit task — do the necessary tidying up afterwards.

What about a response $2x+x^2+1$? This is, arguably, better in the sense that the terms are gathered, but the student here has not *ordered* terms to write their expression in the conventional form[1]. We might say that it is:

mathematically correct, in that it is equal to the question posed;
pedagogically correct, in that the student has done the task required;
aesthetically incorrect, in that there are more conventional ways of writing the answer.

We will not go further into aesthetic correctness here except to point out that it is more difficult, and subjective, than it seems — Table 2 shows two different questions with what most people would agree to be the aesthetically correct answers. Note that the aesthetic answers, while different, are *mathematically* the same, and indeed on a deeper level the questions are the same. Nevertheless, we hope the three-fold classification above is useful.

Table 2. Aesthetically correct answers

Question	Aesthetic answer
Simplify $\frac{x^5-1}{x-1}$	$x^4+x^3+x^2+x+1$
First five terms of Maclaurin series for $\frac{1}{1-x}$	$1+x+x^2+x^3+x^4$

We should note that we have refrained from using the word “simplify” here. The word is ambiguous and indeed it can be used for the opposite mathematical operations. For example, in [33] the word “simplify” is usually taken to mean (e.g. p. 11, Ex 8) “*simplify by removing brackets and collecting like terms*”. (Arguably “removing” should be “expanding” here). But, “simplify” is also later used to implicitly mean *factor and cancel like terms*. For example,

$$\text{p. 139, (77) Simplify} \frac{a^4+a^2b^2+b^4}{a^3-b^3}.$$

[1] We might use the phrase “canonical form”, but this has a technical meaning in computer algebra [14, p. 79].

This is typical[2] of contemporary usage. "Simplify" may mean little more than "do what I've just shown you"[3]: a more refined vocabulary is necessary.

In formal computer science however, for two equivalent expressions A and B, [8] argued that A was simpler than B when "*the length of the description of A is shorter then the length of the description of B*". Applying his formal definitions to binary encodings of the integers and operations $+$, $\times$, $-$ and exponentiation he argued that 2^7 is more complex than 128, but that 2^8 is simpler than 256. However, given two explicit integers n and m "*it is never the case that the algebraic expression $n+m$ is simpler than the integer q equal to $n+m$.*" A restricted version of this, but adequate for our purposes, is implemented in Maple's `simplify(...,size)`.

These issues might, at first, appear utterly trivial. The expert does not worry about such distinctions: a hallmark of their expertise is that they work modulo such "technicalities". But, during elementary mathematical instruction this is *the point of the work.* One application of computer algebra is to automatic computer aided assessment of mathematics and current systems go well beyond multiple choice or similar question types. In particular, students are expected to provide a mathematical expression as their answer and a computer algebra system seeks to establish its properties. On the basis of these properties feedback is provided. In our examples above the teacher might like to say, for example, *"yes, but ... "*. Only if such fine grained distinctions can be made may sufficiently sophisticated feedback be provided. But why is it important to provide such detailed feedback? Is it not sufficient to provide only a binary correct/incorrect outcome, an associated mark and give a student a summary percentage at the end? It is a paradigm in education that "feedback promotes learning". But a closer inspection reveals a much more complex picture. The meta-analysis of [21] examined about 3000 educational studies and found that over one third of feedback interventions *decreased performance*: a counterintuitive and largely ignored outcome. It is not feedback, *per se*, but the nature of the feedback which determines its effectiveness. In particular, feedback which concentrates on specific *task related* features and on how to improve is found to be effective, whereas feedback which focuses on the *self* is detrimental. A low end of test summary mark — hardly a specific form of feedback — may be interpreted as a personal and general comment on the ability of the student, whereas detailed feedback on each task points to where improvement can be made.

[2] The quality and variety of exercises in [33] is, in the opinion of the third author, somewhat better than many current algebra textbooks. C. O. Tuckey was a very well respected teacher, president of the Mathematical Association, author of many books and widely circulated reports (e.g. [34]) into effective teaching. This example is not a personal criticism, but rather an example of typical usage.

[3] This was brought out when the second author taught a summer school of teachers in the French "classes préparatoires" — a system that does not fit into the Bologna framework [4], but is part of the higher education system [3]. The teachers eventually admitted that "simplify" (actually "simplifier", but in this case the English and French words seem to be in close correspondence) meant "give me what I expect".

3 Theoretical Models of Computer Algebra

We have seen there is a big difference between pedagogically correct and mathematically correct, so it might be thought that it might be difficult to reconcile the two, but it is our thesis that it is possible to make the differences amenable to formal treatment. To do this we need to set up some formalism.

While computer algebra has a long history (early approaches include [19,22]), it was largely aimed at supporting specific calculations, and theoretical underpinnings were slower to emerge. The most relevant for our point of view is the "universal algebra" approach underpinning Axiom [18] and Magma [7].

This is generally considered via the 'multi-sorted approach' [32], though in fact algebra systems in practice use an 'order-sorted' approach [17], for the reasons given in [15]. Such a typed approach is very relevant for mathematics as, although the Zermelo-Frankel formalisation of mathematics is untyped, most mathematics in practice is typed, and the mathematical operations have a type structure. In this approach we introduce various operators, so a trivial construction of the integers would introduce `pred` and `succ` operators to define predecessors and successors of numbers, and we introduce axioms, such as the axiom

$$\texttt{pred}(\texttt{succ}(z))=z. \tag{2}$$

We then let $\equiv$ be the congruential closure of our given axioms, such as (2), i.e. the smallest relation containing the axioms and satisfying

R for all t, $t \equiv t$;
S if $t_1 \equiv t_2$, then $t_2 \equiv t_1$,
T if $t_1 \equiv t_2$ and $t_2 \equiv t_3$, then $t_1 \equiv t_3$;
C if $t_1 \equiv t_2$, then

$$f(u_1,\ldots,u_{k-1},t_1,u_{k+1},\ldots,u_n) \equiv f(u_1,\ldots,u_{k-1},t_2,u_{k+1},\ldots,u_n),$$

where f is any n-ary operator.

Because of condition **C**, the operators are well-defined on the equivalence classes of $\equiv$. $\equiv$ is then said to be a **congruence**, and the corresponding logical system "equality up to $\equiv$" is said to be **congruential** [12, Definition 1]. Note that we are *not* saying that computer algebra systems *are* implemented this way, merely that one can formalise what they are doing in this structure.

Hence the first question one can ask of a computer algebra system is the following.

Question 1. Which axioms generate the congruence `=`?

Most algebra systems do not answer this question in full generality, with fully-typed systems such as Axiom and Magma coming the nearest, in the sense that one can inspect the code for that component of $=$ acting on a particular sort.

As far as the authors[4] can determine, for Maple acting explicitly[5] on Laurent polynomial objects built up from the integers and variables (or expressions which behave like variables) with `+`, `-`, `*` and raising to explicit integer[6] powers, the following axioms generate Maple's equality.

1. Associativity of addition (essentially by regarding it as an n-ary operation).
2. Associativity of multiplication (also by regarding it as an n-ary operation).
3. Commutativity of addition.
4. Commutativity of multiplication.
5. Arithmetic evaluations on integer sub-expressions.
6. Collection of $mZ + nZ$ into $(m + n)Z$, where m, n are integers.
7. Replacing $m(Z_1+Z_2)$ by mZ_1+mZ_2 *where*[7] we have precisely a two-element product.
8. Collection of $Z^m Z^n$ into $Z^{(m+n)}$, where m, n are integers (and possibly not explicit if they are 1, though they are stored as such internally).
9. Suppression of $+0$.
10. Suppression of $*1$.
11. Replacing Z^0 by 1.

We note that this does *not* include the general distributive law for multiplication (or its corollary, the expansion of powers). Other systems may vary here.

Where practicable, algebra systems go further, and wish to create a **canonical** representation [14, p. 79], i.e. reduce every element of an equivalence class to one particular representation. In the case just mentioned above, Maple does this by applying rules (5–11) in the left-to-right sense, and storing the components of sums and products in a unique order determined by Maple's internal hash coding system [10]. It is this internal order that causes *apparently* strange results, e.g.

$$\text{simplify}\left(\frac{x^{105}-1}{x-1}\right) = 1 + x + x^{88} + x^{89} + x^{104} + x^{90} + \cdots .$$

Even apart from this problem, asking that a student return the canonical representation is normally too strong. For example, only one of these two expressions can be canonical:

$$\sin x \cos x \text{ or } \cos x \sin x, \tag{3}$$

but it would be a rare teacher who marked one right and the other wrong.

[4] They are grateful to Jacques Carette for his assistance here, but the authors bear the responsibility for any misconceptions.

[5] That is to say, where every sub-expression in the expression is built up this way, rather than by having more complicated operators 'cancel'.

[6] Including negative integers. Note that, while Maple *prints* `x^(-2)` as $\frac{1}{x^2}$, it is in fact stored as x^{-2}. Similarly, x is stored as x^1.

[7] This caveat means that Maple's `=` relation is not actually a congruence.

4 Theoretical Models of Computer-Assisted Pedagogy

This section contains examples, which commonly occur in pedagogy, of senses in which two expressions are the same. It is rare that when assessing a question the teacher considers a single property. In fact, they make a number of separate judgements and construct feedback on the basis of these multiple outcomes. Hence, in a particular situation, the teacher might wish to consider a number of comparisons to build the appropriate feedback. Exactly what outcomes, i.e. mark, feedback and note, to assign is highly context dependent. For example, successfully establishing equivalence to an incorrect answer known to arise from a common misconception may result in no mark, but helpful feedback. Typically, the first test is to establish that the student's answer is "equal" to the correct answer given by the teacher.

There are many senses in which two expressions are considered "equal". We shall describe some of these now, from the most restrictive to the most liberal senses. We provide examples where establishing this sense of equality is a crucial component in the assessment of a mathematical question. However, we do not comment at this stage of the technical feasibility or the efficiency (i.e. computational cost) of doing so.

$==$ The most restrictive sense of equality is *absolutely identical expressions.* For example, the teacher may want exactly $x^2 + 2x + 1$. The order of the terms here is a key component. This kind of equality is closely related to the equality of the parse trees representing the two expressions.

$=_{AC}$ The next notion of equality is that up to commutativity and associativity of the basic arithmetic operations. However, the basic arithmetical operations are assumed to be *nouns*. This means they *represent* the operation, but do not perform the calculation. Hence, $2x + y =_{AC} y + 2x$ but $x + x + y \neq_{AC} 2x + y$. This is a very useful test for checking that an answer is the "same" but "simplified". Since distribution amounts to *doing* multiplication we have $2(x + 1) \neq_{AC} 2x + 2$.

$=_{\text{ext}}$ Extensional equivalence is perhaps the most common notion of equivalence. Take two expressions ex_1 and ex_2, which might contain multi-variables, be an equation, list, set, matrix, etc. If when values are assigned to the variables (from some agreed sets) they always evaluate identically then ex_1 and ex_2 are extensionally equivalent. This notion of equivalence carries over to equations and inequalities. From a technical point of view, we cannot simply evaluate an expression over an infinite set, such as the real numbers. Hence, the starting point for CAS-supported CAA was to evaluate the difference of two expressions symbolically and look for a zero result. There are significant difficulties in establishing extensional equivalence for complex (inverse) trigonometrical expressions [5], and even the apparently trivial

$$\log\left(\frac{1}{x}\right) = -\log x \tag{4}$$

is true everywhere except on a set of measure zero (the branch cut for log, traditionally $(-\infty, 0)$). It is also not immediately clear how in practice to establish whether two *equations* are extensionally equivalent. Systems of inequalities are similarly difficult.

$=_\alpha$ Consider the following question.

> A rectangle has length 8cm greater than its width. If it has an area of 33cm^2, write down an equation which relates the side lengths to the area of the rectangle.

The kind of answer the teacher is looking for is $x(x+8) = 33$, or $l(l-8) = 33$, or indeed $l^2 + 8l - 33 = 0$, or .... One might argue that the phrase "use x to denote the length of the shortest side" could be used here to reduce the technical difficulty of automatically assessing the answer. However, this might significantly reduce the level of difficulty of the problem for the student by making a crucial choice in precisely the modeling step the problem is designed to assess. Given two expressions ex_1 and ex_2, we need to establish whether there exists a substitution of the variables of ex_2 into ex_1 which renders ex_1 extensionally (or whatever other kind of equivalence we are asking for) equivalent to ex_2. This is the idea of α-equivalence, denoted $\equiv_\alpha$ in [2, Definition 2.1.11].

This last notion of equality, i.e. modulo variable names used, might be extended to other kinds of equalities. Indeed, we might well want to know whether there exists a substitution of the variables of ex_2 into ex_1 which renders the new parse tree for the substituted version of ex_1 identical to the parse tree for ex_2.

5 Unifying the Approaches

Let us assume that we *are* trying to use a computer algebra system to get close to *intensional* equivalence, which in the pedagogic context could be described as "does it *mean* the right thing?".

Notation 1. *Let us suppose we have a question, to which the teacher has supplied a formula as the answer, f_T, and the pupil has supplied an answer f_P.*

There are various questions we might ask.

1. Is f_P *identical* to f_T, written $f_P == f_T$? We should note that this question is not trivial to answer, even at the level of MathML-Presentation [11], however, we will assume that it is answerable. If so, the answer is presumably mathematically, pedagogically and even aesthetically correct.
2. Is f_P *mathematically equal* to f_T, at least as far as our algebra system can deduce it, written $f_P =_{\text{CAS}} f_T$? Note that $=_{\text{CAS}}$ may well have to be more sophisticated than just $=_{\text{Maxima}}$ or $=_{\text{Maple}}$. To get answer 5 for Table 1 correct, we need to use $=_{\text{Maxima:}\texttt{trigexpand}}$ and so on. If this is the case, the student has produced a mathematically, even if not pedagogically, correct answer. If not, there are then logically two possibilities.

- The algebra system is wrong (or at least inadequate). This is one of those "should not happen" cases, but might, particularly if the problem-setter (e.g. teacher) has allowed a more powerful input syntax than was intended, as might happen if a (troublesome) pupil answered (1) with

$$x^2 + \left(\max_{n \in \mathbf{N}} \exists x, y, z \in \mathbf{N}^* x^n + y^n = z^n \right) x + 1. \tag{5}$$

  Of course, it is only since the Wiles–Taylor proof that we have known that this was well-defined, never mind correct. In this category also belong all the "computer algebra is undecidable" paradoxes [26], which in practice do not crop up, and can generally be excluded syntactically — after all what business has a pupil got using syntax like (5) at this level?
- The pupil's answer is definitely wrong. This system can do no more to help, and we may wish to look at "buggy rules" (see [23]) or other techniques to determine how much partial credit to allow.

So we are left with a mathematically correct answer, and the question is "how many marks, if any, should be allocated?" (recalling that, in Table 1, one correct answer got no marks), with a supplementary of "what feedback do I need to give?". To be concrete, consider the example of Figure 1, for which we have to add more rules to (1–11) above, say the following.

12. $\frac{\mathrm{d}u^n}{\mathrm{d}x} = nu^{n-1}\frac{\mathrm{d}u}{\mathrm{d}x}$.
13. $\frac{\mathrm{d}\sin(g(x))}{\mathrm{d}x} = \cos(g(x))\frac{\mathrm{d}g(x)}{\mathrm{d}x}$.
14. $\frac{\mathrm{d}uv}{\mathrm{d}x} = u\frac{\mathrm{d}v}{\mathrm{d}x} + v\frac{\mathrm{d}u}{\mathrm{d}x}$.
15. $\frac{\mathrm{d}x}{\mathrm{d}x} = 1$.
16. $\frac{\mathrm{d}n}{\mathrm{d}x} = 0$ (n a number).

Furthermore, we will only let these rules act from left to right (matters might be different if we were setting integration problems rather than differentiation ones). Let us classify the rules into three classes.

underlying: those which we believe do not really change the expression in *form* as well as substance, and which "ought" to be part of ==. Call this class $\mathcal{U}$, and the congruence generated by $\mathcal{U}$ $\equiv_{\mathcal{U}}$. In our case, $\mathcal{U}$ would be (1)–(4).

venial: those which the pupil *ought* to have used, and which should not be left in the pupil's answer. Call this class $\mathcal{V}$, and the congruence generated by $\mathcal{U} \cup \mathcal{V}$ $\equiv_{\mathcal{V}}$. In our case, $\mathcal{V}$ would be (5)–(11).

fatal: those which the pupil *had* to apply, and which *must* not be left in the pupil's answer. Call this class $\mathcal{F}$, and the congruence generated by $\mathcal{U} \cup \mathcal{V} \cup \mathcal{F}$ $\equiv_{\mathcal{F}}$. In our case, $\mathcal{F}$ would be (12)–(16).

In fact, $\equiv_{\mathcal{F}}$ should be $=_{\mathrm{CAS}}$, i.e. everything that our algebra system can prove.

Then we can propose the following strategy (Table 3) for our automated marker, depending on the *finest* relation R for which $f_P R f_T$. This leads to the results in Table 4, where we see two differences from Table 1: item 3 is simply marked wrong, rather than being given 0.7, and item 5 is marked wrong, whereas

Table 3. Putative Strategy

Relation	Score	Feedback
$\equiv_{\mathcal{U}}$	1.0	Well done
$\equiv_{\mathcal{V}}$	0.8	OK, but there are better ways of writing it
$\equiv_{\mathcal{F}}$	0.0	You were meant to *do* the differentation
—	0.0	I'm sorry, that's not right

in fact it is right. The first of these is a "buggy rule" issue, as discussed earlier. As regards the second, the problem is that we have not told the system about trigonometric contraction. This involves adding a new rule

17. $\sin x \cos x = \frac{1}{2} \sin 2x$

as well as various rules about fractions, which should pretty certainly be added to class $\mathcal{V}$. Before we can discuss the correct classification of rule 17, we need a digression.

Table 4. Table 1 according to table 3

No.	Student's answer	Score	Feedback
1.	$4\sin 2x \cos 2x$	1	Well done
2.	$\frac{\mathrm{d}\sin^2 2x}{\mathrm{d}x}$	0	*do* the differentation
3.	$2\sin 2x \cos 2x$	0	I'm sorry, that's not right
4.	$2 \times 2\sin 2x \cos 2x$	0.8	better ways of writing it
5.	$2\sin 4x$	0	I'm sorry, that's not right
6.	$2\sin 2x \cos 2x + 2\sin 2x \cos 2x$	0.8	better ways of writing it
7.	$x/4 - \sin(4 * x)/8$	0	I'm sorry, that's not right

6 What Is a 'Right' Answer (Algorithmically)?

This is an important question. If we are following [16] and generating *questions*, we must also generate *answers*, as well as mark schemes on the lines of Table 3. Here we follow the suggestion of [8] and say that a 'right' answer a must be:

(a) equivalent under $\mathcal{U} \cup \mathcal{V} \cup \mathcal{F}$ to the question asked (a is mathematically an answer);
(b) invariant under the application of the rules in $\mathcal{F}$ (a isn't the question restated);
(c) a smallest such member, i.e. there is no a' with $a \equiv_{\mathcal{V}} a'$ and $|a'| < |a|$ for some size measure $|\cdot|$.

Quite what we take as our definition of $|\cdot|$ is not clear, and probably needs further experimentation. For the moment we are taking the number of printed characters in the answer, though a case could certainly be made for including `&InvisibleTimes;` and `&FunctionApplication;` as well. In this context, assuming (17) is not in $\mathcal{F}$, we see that $2\sin 4x$ is 'a' right answer, and indeed 'the' right answer.

6.1 Trigonometric Contraction Revisited

With this preamble, we can now ask about the classification of (17). If we are concerned merely about differentiation, we could class it with $\mathcal{U}$. If we had already taught trigonometric contraction, we could class it in $\mathcal{V}$. Alternatively, we could be more subtle. Suppose we had taught it before, and wanted to give the students a gentle reminder of it. We could define $\equiv_{\mathcal{U}'}$ to be the congruence generated by $\mathcal{U} \cup \{(17)\}$, and use the score and feedback from Table 5. Alternatively, we may have made the point before, and want to reinforce it. Then could define $\equiv_{\mathcal{V}'}$ to be the congruence generated by $\mathcal{U} \cup \mathcal{V} \cup \{(17)\}$.

Table 5. Putative Strategy Refined

Relation	Score	Feedback
$\equiv_{\mathcal{U}}$	1.0	Well done
$\equiv_{\mathcal{U}}{}'$	0.9	Well done, but you forgot about trigonometric contraction
$\equiv_{\mathcal{V}}$	0.8	OK, but there are better ways of writing it
$\equiv_{\mathcal{V}'}$	0.6	You *really* should use trigonometric contraction
$\equiv_{\mathcal{F}}$	0.0	You were meant to *do* the differentation
—	0.0	I'm sorry, that's not right

6.2 (1) Revisited

To resolve (1) in this framework, we would let $\mathcal{U}$ be rules (1)–(4), $\mathcal{V}$ be rules (5)–(11), and $\mathcal{F}$ be the following (interpreted as left→right rules).

18. $Z^n = Z \cdot Z^{n-1}$.
19. n-ary distributive law.

This gives us the results in Table 6.

Table 6. Question 1 according to table 3

No.	Student's answer	Congruence	Score	Feedback
1.	$x^2 + 2x + 1$	$\equiv_{\mathcal{U}}$	1	Well done
2.	$(x + 1)(x + 1)$	$\equiv_{\mathcal{F}}$	0	*do* the expansion
3.	$x^2 + 2x + 2$	—	0	I'm sorry, that's not right
4.	$x^2 + x + x + 1$	$\equiv_{\mathcal{V}}$	0.8	better ways of writing it

6.3 Other Issues

The way systems deal with numbers is also crucial. The representation 0.5 actually means five tenths. Students are apt to write things such as $0.5x^2 + 1/3$, a perfectly accurate representation for $x^2/2 + 1/3$, but one which does not conform to notational conventions. So, the system should establish an equality when floating point numbers are used within expressions. However, 0.33 is often not an acceptable approximation for 1/3. Whether the teacher will reject all expressions containing floating point numbers, or whether they wish to say "yes you are correct, but we don't normally use floats" is a matter of pedagogy and should not be a work around a technical restriction. Some of these issues can be solved in our framework: for example we could have a rule converting decimals into the corresponding rationals, and place it into $\mathcal{U}$ (no penalty), $\mathcal{V}$, or possibly some $\mathcal{V}''$ with a penalty, and feedback, of its own.

Lastly, particularly for assessment of science, we would like to deal with units. Here is is necessary to establish whether the student has the correct value and correct units, or the "correct value" using different but dimensionally consistent units to that of the teacher. This area requires its own reasoning [30], but which could nevertheless be incorporated into this framework by asking which rules (OpenMath Formal Mathematical Properties) were used, and whether their use incurs a penalty.

We highlight some closely related issues. For example, a naïve set is a collection of objects, without duplication. Given the many senses of equality above, it is necessary to be explicit about how the set construction function decides on the equality of two given expressions. Rarely is extensional equivalence actually used. For example, in Maple 9.5 we have the following session:

```
> S:={x^2-1,(x-1)*(x+1)};
```

$$S := \left\{(x-1)(x+1), x^2-1\right\}$$

```
> map(simplify,S);
```

$$S := \left\{x^2-1\right\}$$

When written in different algebraic forms, Maple is happy to tolerate duplicates in sets. The default notion of equality is not that of extensional equivalence. Indeed, for many CAS the notion of equality for the purposes of sets is simply that of identity of internal representations, once any default "simplification" has been done. Here again we would need a set of rules, some underlying (e.g. order of elements in a set doesn't matter), some venial or fatal (e.g. removal of duplicates), depending on the pedagogical point being stressed.

7 Conclusion

This paper has looked at the question "how hard is it to use computer algebra to decide if a 'Calculus 101' answer is correct?", and, we hope, convinced the reader that it is rather harder than it looks. No matter which CAS we use, $=_{\text{CAS}}$ is simultaneously too strong and and too weak for what we want.

Too strong: it may decide that simple restatement of the questions are "correct", because they are algebraically equivalent to the answer.

Too weak: the built-in $=_{\text{CAS}}$ does not apply enough rules, such as trigonometric contraction.

Too coarse: it cannot produce the "mostly right but" answers we have allocated 0.8 to above. Of course, the number of marks is, of course, a matter of taste for each teacher to decide and our somewhat arbitrary allocations should not be taken too seriously.

Too inflexible: the set of rules allowed, and their status within the marking scheme, will vary *during* a single course, never mind between courses.

This is not to say that computer algebra is not useful, and indeed $=_{\mathrm{CAS}}$ will probably be an important *component* of any scheme. But such a scheme will need to have different levels of equality. For simplicity, we have illustrated a linear hierarchy, but in practice one would probably have a lattice of various classes of "venial" rules.

References

1. Ashton, H., Beevers, C.E., Koraninski, A.A., Youngson, M.A.: Incorporating partial credit in computer aided assessment of mathematics in secondary education. British Journal of Educational Technology 37, 93–119 (2006)
2. Barendregt, H.P.: The Lambda Calculus: Its Syntax and Semantics. North-Holland, Amsterdam (1984)
3. Belhoste, B.: Historique des classes préparatoires. Exposé au Colloque de l'UPS (2003), `ftp://trf.education.gouv.fr/pub/edutel/sup/cpge/historique.pdf`
4. Bergen Conference of European Ministers Responsible for Higher Education. The framework of qualifications for the European Higher Education Area (2005), `http://www.bologna-bergen2005.no/EN/BASIC/050520_Framework_qualifications.pdf`
5. Bradford, R.J., Davenport, J.H.: Towards Better Simplification of Elementary Functions. In: Mora, T. (ed.) Proceedings ISSAC 2002, pp. 15–22 (2002)
6. Cajori, F.: A history of mathematical notations. Open Court (1928)
7. Cannon, J., Playoust, C.: An Introduction to MAGMA. Springer, Heidelberg (1997)
8. Carette, J.: Understanding expression simplification. In: Gutierrez, J. (ed.) Proceedings of ISSAC 2004, pp. 72–79 (2004)
9. Chaachoua, H., Nicaud, J.F., Bronner, A., Bouhineau, D.: APLUSIX, a learning environment for algebra, actual use and benefits. In: Proceedings of the International Congress on Mathematics Education (ICME-10), Copenhagen, Denmark (2004)
10. Char, B.W., Geddes, K.O., Gentleman, M.W., Gonnet, G.H.: The Design of MAPLE: A Compact, Portable and Powerful Computer Algebra System. In: van Hulzen, J.A. (ed.) ISSAC 1983 and EUROCAL 1983. LNCS, vol. 162, pp. 101–115. Springer, Heidelberg (1983)
11. World-Wide Web Consortium. Mathematical Markup Language (MathML) Version 2.0, 2nd edn. (2003), `http://www.w3.org/TR/MathML2/`
12. Davenport, J.H.: Equality in computer algebra and beyond. J. Symbolic Comp. 34, 259–270 (2002)
13. Davenport, J.H., Libbrecht, P.: The Freedom to Extend OpenMath and its Utility. Mathematics in Computer Science (to appear, 2009)
14. Davenport, J.H., Siret, Y., Tournier, E.: Computer Algebra, 2nd edn. Academic Press, London (1993)
15. Doye, N.J.: Automated Coercion for Axiom. In: Dooley, S. (ed.) Proceedings ISSAC 1999, pp. 229–235 (1999)

16. Gage, M., Pizer, A., Roth, V.: WeBWorK: Generating, delivering, and checking math homework via the Internet. In: Proc. ICTM2 international congress for teaching of mathematics at the undergraduate level (2002), `http://www.math.uoc.gr/~ictm2/Proceedings/pap189.pdf`
17. Goguen, J.A., Meseguer, J.: Order-sorted Algebra I: Equational deduction for multiple inheritance, polymorphism and partial operations. Theor. Comp. Sci. 105, 217–293 (1992)
18. Jenks, R.D., Sutor, R.S.: AXIOM: The Scientific Computation System. Springer, Heidelberg (1992)
19. Kahrimanian, H.G.: Analytic differentiation by a digital computer. M.A. Thesis, Temple University (1953)
20. Klai, S., Kolokolnikov, T., Van den Bergh, N.: Using Maple and the web to grade mathematics tests. In: Proceedings of the International Workshop on Advanced Learning Technologies, Palmerston North, New Zealand, December 4–6 (2000)
21. Kluger, A.N., DeNisi, A.: Effects of feedback intervention on performance: A historical review, a meta-analysis, and a preliminary feedback intervention theory. Psychological Bulletin 119(2), 254–284 (1996)
22. Nolan, J.: Analytic differentiation on a digital computer. M.A. Thesis, M.I.T. (1953)
23. O'Shea, T.: A self improving quadratic tutor. In: Sleeman, D., Brown, J.S. (eds.) Intelligent Tutoring Systems, ch. 13, pp. 309–336. Kluwer, Dordrecht (1982)
24. Ramsden, P.: Fresh Questions, Free Expressions: METRIC's Web-based Self-test Exercises. Maths Stats and OR Network online CAA series (June 2004), `http://www.mathstore.ac.uk/repository/mathscaa_jun2004.pdf`
25. Recorde, R.: The Whetstone of Witte. J. Kyngstone, London (1557)
26. Richardson, D.: Some Unsolvable Problems Involving Elementary Functions of a Real Variable. Journal of Symbolic Logic 33, 514–520 (1968)
27. Sangwin, C.J.: STACK: making many fine judgements rapidly. In: CAME (2007)
28. Sangwin, C.J.: What is a Mathematical Question? In: Proceedings of the JEM conference, Lisbon (Feburary 2007)
29. Sangwin, C.J., Ramsden, P.: Linear syntax for communicating elementary mathematics. Journal of Symbolic Computation 42(9), 902–934 (2007)
30. Stratford, J.D., Davenport, J.H.: Unit Knowledge Management. In: Autexier, S., Campbell, J., Rubio, J., Sorge, V., Suzuki, M., Wiedijk, F. (eds.) AISC 2008, Calculemus 2008, and MKM 2008. LNCS, vol. 5144, pp. 382–397. Springer, Heidelberg (2008)
31. Strickland, N.: Alice interactive mathematics. MSOR Connections 2(1), 27–30 (2002), `http://ltsn.mathstore.ac.uk/newsletter/feb2002/pdf/aim.pdf`
32. Thatcher, J.W., Wagner, E.G., Wright, J.B.: Data Type Specification: Parameterization and the Power of Specification Techniques. ACM TOPLAS 4, 711–732 (1982)
33. Tuckey, C.O.: Examples in Algebra. Bell & Sons, London (1904)
34. Tuckey, C.O.: The teaching of algebra in schools. A Report for the Mathematical Association. G. Bell & Sons (1934)