equivcalc.expeditions
created 07-10-2007
revised last 06-12-2009

Three Expeditions in Search of Hidden Treasure: New Proofs in Equivalential Calculus

Larry Wos

Mathematics and Computer Science Division
Argonne National Laboratory
Argonne, IL 60439
wos@mcs.anl.gov

Why search for treasure? If the decision is to make such a search, what kind of treasure merits the time and effort? Once the type of treasure has been chosen, how do you decide where to search? And, do you need to be an expert to find treasure? I'll answer this last question immediately: No, you need not be an expert; in fact, if you become intrigued by what you read here, you may be able to find more treasure without mastering the implied subtleties.

Here I tell a story of three such searches, three expeditions in search of hidden treasure. You can view this narrative as a notebook, one written by a researcher who relies very heavily on a charming and powerful computer program, namely, OTTER, designed and developed by William McCune. (Although I write about what I was able to find with OTTER, similar treasure can be found with some other automated reasoning program, but I think the program had best offer a wide variety of strategies.) You will learn why I made these expeditions, what the nature of the treasure is, and where the expeditions took me. If all goes as implicitly planned, perhaps one of you will continue the search--especially if you like puzzles or games! Further, if my friend and colleague Ross Overbeek is right, a thorough study of how the journeys proceeded will eventually lead to significant insights, some focusing on finding proofs thought to be out of reach.

If you like improvements in numbers as I do--batting averages, track and field results, swimming times--you might find more than piquant the results obtained and cited here. If you enjoy the conquering of obstacles and, even more, the methods that were used, here also you will find excitement. To encourage you to seek treasure of the kind in focus here, I shall include a sample input file and a sample proof that can provide you with a starting point.

Raison d'etre

In addition to hoping to motivate readers of this story to join in my search for treasure, I am writing this as a record of the events that led to the mining of the treasure. As many know, sometimes more important than the actual treasure is the excitement of the hunt, and sometimes more important is the nature of the hunt. I suspect that, for many who will read this, the nature will in the long run prove more interesting and more valuable. Indeed, if my colleague Ross Overbeek has his way, this notebook (as well as others I am writing) will shed much light on the properties of the unbelievably huge set of conclusions that can be drawn from the simplest of hypotheses. For example--although not everyone's cup of tea, but take a sip

This work was supported in part by the U.S. Department of Energy, under Contract DE-AC02-06CH11357.

anyway--there are various fields of logic and various fields of mathematics in which a single formula, or equation, suffices to study the whole. Even if the tea being served does not offer you a fine flavor, you may still enjoy this narrative, especially if you like various computer programs.

Indeed, Bill's program OTTER played a vital role in all three expeditions to be featured. That program automates logical reasoning, the drawing of conclusions from given hypotheses in a manner that is flawless. With the assistance of OTTER, you need not understand the fine details regarding the theory underlying this story, for that program will apply an appropriate inference rule for drawing conclusions. Instead, as (for example) in car-racing, you drive the car with no requirement of mastering the elements of engineering. Put another way, you can win many poker tournaments without having studied mathematics to any extent. Therefore, I invite you to share the tactics and strategy I employed to reap the rewards featured here. Even better, I invite you to extend what you find in this article; yes, although the mine is deep, I do not believe the vein has run out.

One such mine is in the area of logic known as equivalential calculus. This area of logic admits single axioms, single formulas from which (generally speaking) the entire set of theorems can be deduced. You can casually think of this area of logic as concerned with equivalence or equality. You perhaps have seen the properties of reflexivity, symmetry, and transitivity, represented here by the following, where the function e denotes equivalence and the predicate P denotes provability.

```
P(e(x,x)).  %  reflexivity
P(e(e(x,y),e(y,x))).  %  symmetry
P(e(e(x,y),e(e(y,z),e(x,z)))).  %  transitivity
```

Since equivalential calculus has the cited concern (of equivalence), the three clauses just given (which, in that form, are called clauses) should, intuitively, provide a complete axiomatization. Because these clauses, taken together, constitute an axiom system, a path to any of the more interesting theorems must exist starting with the three and applying an appropriate inference rule, namely, condensed detachment (defined almost immediately). In fact, such is the case.

Condensed detachment is the following, where "-" is interpreted as logical not and " |" as logical or.

```
-P(e(x,y)) | -P(x) | P(y).
```

Perhaps unexpected, the first of the three clauses (axioms)--reflexivity--is deducible from the other two. In other words, the second and third clauses (axioms) provide a complete axiomatization, a 2-basis.

Some of you may be trying, at this very moment, to prove reflexivity dependent on symmetry and transitivity; others may even doubt that it is true. I was unaware of this dependence decades ago when I wrote a paper on logic. Robert (Bob) Veroff is the person who told me about the dependence. Now, if you are working on the problem and do not wish to see a proof, you had best pause here. Indeed, I am about to give a proof, a short proof, of the dependence. When and if you read the proof, do not be discouraged if you find it far from transparent. I now give the proof, relying on the use of condensed detachment (captured by the use of an inference rule called hyperresolution and the appropriate clause), that reflexivity is dependent on the pair, symmetry and transitivity.

```
----- Otter 3.3g-work, Jan 2005 -----
The process was started by wos on lemma.mcs.anl.gov,
Tue Jul 31 08:49:18 2007
The command was "otter".  The process ID is 1035.

----> UNIT CONFLICT at   0.02 sec ----> 152 [binary,151.1,109.1] $ANSWER(reflex).
```

Length of proof is 2. Level of proof is 2.

---------------- PROOF ----------------

```
90 [] -P(e(x,y)) | -P(x) |P(y).
93 [] P(e(e(x,y),e(y,x))).
94 [] P(e(e(x,y),e(e(y,z),e(x,z)))).
109 [] -P(e(a,a)) | $ANSWER(reflex).
127 [hyper,90,93,94] P(e(e(e(x,y),e(z,y)),e(z,x))).
151 [hyper,90,127,93] P(e(x,x)).
```

As for a specific and charming example of a single axiom for EC (equivalential calculus), the following formula, known as XCB, provides yet another key item for this story being told.

```
P(e(x,e(e(e(x,y),e(z,y)),z))). % XCB
```

The formula XCB is the last of its kind, sad to say. In particular, no other formula of this length will ever be found that can serve as a single axiom for equivalential calculus. Its power, its status of being a single axiom, was proved on April 13, 2002. That occurrence concluded an intense study by my esteemed colleague D. (Ted) Ulrich and me. Without an automated reasoning program offering OTTER's numerous options--and few programs offer a comparable set of strategies for restricting and directing the reasoning--the question of whether XCB is a single axiom would, I strongly suspect, still be open. With the assistance of that tireless and flawless reasoning program, the final and fourteenth shortest single axiom for EC was found. (No axiom exists, for this area of logic, that, ignoring the predicate symbol, relies on strictly fewer than eleven symbols.)

The type of treasure represented by the answering of the XCB open question is hard to match and strikingly different from the treasure in focus here. In order to understand the nature of the object of the three expeditions to be described, just a bit more is needed. Then it will be time for detailing each of the three journeys and the treasure that was found.

To prove that XCB is a single axiom for equivalential calculus, Ulrich and I chose a natural target. In particular, we chose to show that, from XCB, a proof could be completed that derives some known basis. So, as targets, we focused on the cited 2-basis as well as on thirteen known shortest single axioms, the following.

```
P(e(e(x,y),e(e(z,y),e(x,z)))). % P1_YQL
P(e(e(x,y),e(e(x,z),e(z,y)))). % P2_YQF
P(e(e(x,y),e(e(z,x),e(y,z)))). % P3_YQJ
P(e(e(e(x,y),z),e(y,e(z,x)))). % P4_UM
P(e(x,e(e(y,e(x,z)),e(z,y)))). % P5_XGF
P(e(e(x,e(y,z)),e(z,e(x,y)))). % P7_WN
P(e(e(x,y),e(z,e(e(y,z),x)))). % P8_YRM
P(e(e(x,y),e(z,e(e(z,y),x)))). % P9_YRO
P(e(e(e(x,e(y,z)),z),e(y,x))). % PYO
P(e(e(e(x,e(y,z)),y),e(z,x))). % PYM
P(e(x,e(e(y,e(z,x)),e(z,y)))). % XGK
P(e(x,e(e(y,z),e(e(x,z),y)))). % XHK
P(e(x,e(e(y,z),e(e(z,x),y)))). % XHN
```

(For the scholar, I now in effect quote from a recent e-mail from Ulrich. In brief, the names used here are due to John Kalman, the last letter in each name of a formula is used to specify the exact order in which the sentential variables occurring in the formula in question occur, while the earlier letter or letters specify its bracket type or resonator class. Full details are given on Ulrich's own web site at http://web.ics.purdue.edu/~dulrich/TRIAL-Kalman'sNamesForThe11-symbolEQtheses.htm.) If OTTER, starting with XCB as the sole hypothesis, completed a proof whose last line was one of the given thirteen, or completed a proof deriving both transitivity and symmetry, then Ulrich and I would have shown XCB to be a single axiom for EC. As it turned out, a number of relevant proofs were eventually completed.

The natural question to ask concerns the means for drawing conclusions to be used in a possible proof, whether by a person or by a computer program. As mentioned earlier, the means, the inference rule, is the following, known as condensed detachment, where "-" is interpreted as logical not and " |" as logical or.

-P(e(x,y)) | -P(x) | P(y).

Although this rule can be applied by hand, fortunately you need not add this skill to your collection. Instead, you can rely on OTTER. In fact, you need not grasp the subtleties of this rule if and when you attempt to extend the results given in this story. You can hunt for treasure of the type offered here by applying the methods to be discussed, or by inventing your own.

By this time, your curiosity may have grown exponentially regarding the nature of the treasure. The treasure sought in each of the three expeditions that are central to this story consists of various short proofs. (Short proofs were of interest throughout the decades to such as C. A. Meredith, A. N. Prior, I. Thomas, and, discovered relatively recently, to Hilbert; indeed, still today, the interest in proof shortening is of concern to various logicians and mathematicians.) Yes, a so-called first proof of some thought-to-be true statement is the usual goal in logic and in mathematics, as well as in other fields. But, just as some seek a shorter path to travel on the way home, and some seek a more efficient algorithm for performing some task, throughout the decades some researchers have sought the shortest (or at least a short) proof. Clearly, the seeking of shorter and ever shorter proofs fascinates me. That activity has dominated my research in the past few years. Perhaps disappointing to some, yet intriguing to others, no practical algorithm exists for finding shorter and ever shorter proofs. Also, even with the assistance of a program of OTTER's type, much advice is crucial, advice about where to look, where not to look, which deductions are most appealing, and the like. And advice giving is more subtle than it might appear.

For example--and perhaps far from obvious--if you do not advise the program that a particular deduction is most appealing, its complexity (measured in symbol count) may prevent the program from ever concentrating on it. On the other hand, if you advise the program to concentrate on some specific formula or equation, if and when it is deduced, that formula or equation may lead the program into a cul de sac. Such an item might cause the program to complete a proof of say 30 steps, whereas avoiding its consideration might enable the program to find a 25-step proof.

At this point, I invite you to share my excitement as I recount three expeditions, each designed to find (for me, and perhaps for others) great treasure. If the treasure does not dazzle you, perhaps the various methods and approaches will. For the smallest of tastes, sometimes the key was to wait, and wait for a long, long time--for far more than 100 CPU-hours on a Linux workstation.

The First Expedition

Some background and stage setting are in order. Otherwise, the picture would be more than incomplete. Indeed, my intent (as you can guess) is to spur various individuals to conduct research, extend research, and, at least, delve into new and exciting areas.

In what ways did my experiences, alone or with another researcher, prompt me to make the first expedition? First, for years, as early as 1992, I have been quite consumed by the seeking of shorter proofs, especially those in the published literature. I suggest that at least a few of you would find it exhilarating to find and offer a proof of some result whose length measured in deduced steps, is strictly shorter than the result papers and books offer. Just for clarity, you are not allowed to leave out steps; indeed, you are asked to supply the inference rule or rules, and your proof must give the history of each deduced step. This requirement is more typical of many areas of logic than it is of various areas of mathematics.

The second force that motivated me was our success, that of Ulrich and me, in proving XCB to be a single axiom for equivalential calculus. As noted earlier, we chose as targets various known single axioms as well as the 2-basis consisting of symmetry and transitivity. Our success included a number of proofs of the desired type, although rather long ones when compared to the results I give when Expedition Three is the focus. Of course, far more important than shortening the proofs was the demonstration that the formula XCB is an axiom, a shortest single axiom. You might naturally wonder about some of the aspects of that success.

Certainly a key factor was Ulrich's success with proving (on April 6, 2002) one of the fifteen 7-symbol theorems with XCB as the sole hypothesis. An additional key factor was access to McCune's OTTER and, of course, its options for types of search (which will be briefly addressed at this time) and for advice taking. At the simplest level, OTTER offers two types of search: that of level saturation and that of complexity preference. Regarding the former, an input clause has by definition level 0; a (particular occurrence of a) deduced clause has a level one greater than the maximum of the levels of its parents. With level saturation, within the limits placed on the search by various values assigned to parameters, the program deduces all clauses of level 1, then of level 2, then of level 3, and the like. Intuitively, children are first produced, then grandchildren, then great grandchildren, and the like.

The use of level saturation presents a formidable obstacle. Specifically, in the majority of cases, the levels grow in size very rapidly, which prevents the program from exploring levels much greater than, say, 14.

In contrast, with complexity preference, the program does not consider the clauses in the order they are retained. Rather, the choice for the next clause on which to focus for inference-rule application is based on the program's notion of simplest. If no guidance is given, simpler is measured strictly in terms of symbol count. However, as implied, the user can include various items (weight templates or resonators, for example) that override this measure, informing the program, for example, that some clause is to be treated as simple even though it may rely on many, many symbols.

Ulrich's success of April 6 suggested to me (because of my many, many experiments throughout the years) that OTTER be added to the team. Both a level saturation and a complexity preference search were initiated in separate runs. Each yielded useful proofs, proofs of more of the fifteen 7-symbol formulas. As expected, the level-saturation approach took a great deal of time, four days in real time, but it did yield the key bit of information. (I almost terminated the level-saturation run after two real days because nothing was happening; what an unfortunate choice that would have turned out to be!) The move that opened the door that had been locked for so many decades was to rely on the proof steps of the various proofs of the 7-symbol formulas. For the historian, Robert (Bob) Veroff played a key role in that his superb hints strategy was employed; indeed, the proof steps were adjoined as hints, which I shall briefly explain now.

Hints are not treated as being true or false. Instead, hints are used by OTTER, or by any other automated reasoning program offering that capability, to guide the program in its choice of where next to focus, among other uses. Yes, hints are used for advice giving, for example, to inform the program that a formula or equation, though appearing to be complex, is to be considered simple. To answer a question that would occur to one very familiar with automated reasoning, the hints strategy, when compared to the resonance strategy, requires far less time to rely on in OTTER. Neither is a substitute for the other. Also included as hints were formulas corresponding to a proof Branden Fitelson had found of reflexivity, derived from XCB. (Before the Fitelson proof of reflexivity was offered, history requires noting that some had thought they were on the way to proving XCB too weak to be a single axiom; I certainly, at one time, was such a person.)

Everything went according to plan, or hope. The various 7-symbol formulas, including symmetry, were quickly proved, as well as reflexivity. OTTER completed a 61-step level-20 proof of symmetry, and also completed a 71-step level-23 proof of transitivity. Yes, the 2-basis was proved before any of the known single axioms was deduced, with a proof of length 71 and level 23. In addition, the program completed proofs of P1_YQL, P2_YQF, P3_YQJ, P4_UM, P5_XGF, P7_WN, P8_YRM, P9_YRO, XGK, PYM, PYO, XHK, XHN, and Wajsberg_5.

P(e(e(x,e(y,z)),e(e(y,e(u,z)),e(u,x)))). % Bryman P(e(e(x,e(y,z)),e(e(y,e(z,u)),e(u,x)))). % Lukasiewicz 1 P(e(e(u,e(x,e(y,z))),e(e(x,y),e(z,u)))). % Lukasiewicz 2 P(e(e(x,e(y,z)),e(e(x,e(z,u)),e(u,y)))). % Sobocinski 1 P(e(e(x,e(y,z)),e(e(x,e(u,z)),e(u,y)))). % Sobocinski 2 P(e(e(e(x,e(y,z)),e(e(z,u),u)),e(x,y))). % Wajsberg_4 P(e(e(e(e(x,y),z),u),e(u,e(x,e(y,z))))). % Wajsberg_5

So, in effect, additional targets were offered in the context of proof shortening. (In the Epilogue, I supply, thanks to Ulrich, a bit of charming history concerning single axioms; there I also offer a challenge you might find intriguing.)

Ulrich was--and indeed still is--a motivating force in these expeditions of proof shortening. He is indeed interested in proof shortening. His successes in proof shortening brought me pleasure. He can, with his knowledge and his intuition and his own (interactive) program, see places where steps can be saved; I do not have that ability. Ulrich was as eager as I to find a far shorter proof establishing XCB to be a single axiom for EC.

If memory serves, I believe that Ulrich preferred to focus mostly on the 2-basis as the target. However, because of OTTER's nature, no need existed for focusing exclusively on the 2-basis. Indeed, this program permits the researcher to seek in one run many proofs of many so-called targets at the same time, if that is the choice, as well as many proofs of a single target. Therefore, the goal was to simultaneously seek shorter and ever shorter proofs of the 2-basis, as well as those for each of the other known shortest single axioms, thirteen in number. Immediately you will see how the ability to find many proofs of the same result in one run comes into play.

One of the features of OTTER is the choice of whether to use a procedure called ancestor subsumption. When this procedure is invoked (is set), the program, upon deducing a second copy of a conclusion already retained, will compare the two proofs that deduce the conclusion in the context of proof length (number of deduced steps). With ancestor subsumption, the shorter proof is the one that is kept in focus. In other words, if the second proof is strictly shorter than the first, then that proof is the one that is actively maintained. A natural conjecture asserts that the use of this procedure is just what is needed. Although intuitively reasonable, such is not the case, as the following shows.

First, imagine that the goal is to prove both symmetry and transitivity. Then imagine that the program finds an 18-step proof of the first and a 20-step proof of the second. If the proofs have sixteen deduced steps in common, then the length of the proof of the corresponding 2-basis is 16+2+4 = 22. If the program continued and found a 15-step proof of symmetry and a 17-step proof of transitivity, and if ancestor subsumption were in use, these new proofs would take preference. If this new pair of proofs had but three steps in common, then taken together, the program would present a proof of the 2-basis of length 29, 3+12+14. Finally, if the second pair of proofs was found before the first proof of transitivity was completed, the program would have no interest in a 20-step proof of transitivity. Therefore, in the given case, the 22-step proof of the 2-basis would never be offered. By the way, you can construct a somewhat similar example when the target is a single formula or equation. At the moment, I leave this task to you--if you enjoy challenges or puzzles.

Despite the possibility of getting into trouble (of the type just illustrated) when using ancestor subsumption, I used it and used it heavily in the pursuit of a much shorter proof showing XCB to be a single axiom. And the first expedition had begun, and begun in earnest. But far more would be needed if a fine prize was to be won, a proof of length far less than 71.

Rather than a detailed account, I highlight here some of the key procedures that led to finding the sought-after treasure. That which requires the least explanation concerns hint replacement. When ancestor subsumption succeeds, often the program returns a number of proofs of the same target. They, as earlier remarks justify, are not always smaller in length and still smaller. However, if and when a shorter proof is found, you take its proof steps and, in the next run, replace the hints that were used by hints corresponding to the steps of the new and shorter proof. In a manner of speaking, you are enabling the program to learn about better, more effective sets of hints. Use of the new set of hints, smaller in number than those used to (in effect) find them, will typically lead to the program's either finding the so-called new proof again or, one hopes, finding a still shorter proof. Iteration is the key move, using smaller and still smaller sets of hints.

Next, when the combination of ancestor subsumption and hint replacement stopped showing progress, we turned to an odd use of demodulation. You might keep in mind that, ordinarily, demodulation is used for simplification and canonicalization. In the use now in focus, however, it is used to block the retention of one or more formulas that the user classifies as unwanted. Such clauses are demodulated to junk, a constant used to mean unwanted, and propagated with the appropriate demodulators to enable OTTER to purge the clause. The following list gives the rudiments of what is needed, in addition to including a demodulator that purges upon generation an unwanted clause.

```
list(demodulators).
(e(x,junk) = junk).
(e(junk,x) = junk).
(P(junk) = $T).
end_of_list.
```

Sometimes some formula or equation, when present in a proof, results in a proof longer than necessary. In an extreme case, the program can find a proof all of whose steps are among those of an earlier proof that is one step longer. What has changed is the history, the parentage of a clause. To put it in general terms, two parents (say in the animal world) can have many children; in many logics, for example, one child can have many pairs of parents. With condensed detachment, such is indeed the case, which is one reason why demodulation blocking can work. Two choices exist with this so-called demodulation blocking. You can more or less force the program to consider only the clauses in the proof in hand. Instead, you can give the program additional latitude, allowing it to bring into play clauses not in the proof in hand. During this first expedition, demodulation blocking was often used, in addition to ancestor subsumption and hint replacement.

Next in focus in the pursuit of treasure in the form of yet shorter proofs is a more recent strategy, that of cramming. In contrast to the hints strategy or the resonance strategy, in the cramming strategy items are added that do have a truth value, namely, true. In order to set the stage for its introduction, some motivation is merited. If you return to the example of longer and shorter proofs in the context of the 2-basis, you will again see that the proof of such benefits from the two subproofs (of symmetry and transitivity) sharing as many steps in common as possible. In a way, when a formula or equation participates in two proofs that are each subproofs of a total proof, that formula or equation is doing double duty. If a formula or equation is a parent of three or more later deduced steps, you might say that such a formula is doing triple duty or more. So the task at hand was to devise a method, or strategy, that forces various items to do double duty, triple duty, or more. Put another way, if you have in hand a proof of, say, symmetry and wish to complete a proof of transitivity that relies on many of the proof steps of symmetry, you might wonder how to have this happen.

The cramming strategy was born. It works in the following way. You place the (deduced) proof steps of the proof of symmetry in the input as so-to-speak added axioms or lemmas. In the case of OTTER, you place them in list(sos) so they will be able to initiate applications of the chosen inference rule or rules. You then instruct the program to focus on these added items before it focuses on any newly deduced and retained conclusions. With OTTER, you include the command set(input_sos_first). You can instead choose a level-saturation search, with the command set(sos_queue). That command will, as discussed earlier, first focus on all of the input clauses in list(sos) with the goal of deducing level-1 clauses before deducing level-2 clauses.

To provide more insight (for the reader) into how cramming works and what can happen, the program might deduce and retain a conclusion C such that the proof relies on a number of items that were added to the input from an earlier proof of, say, B. If the program finds a second, quite different proof of C, but of the same length as the first proof, then even with the use of ancestor subsumption, the first proof will be the one that stays in focus. In the case that could occur to you, B is symmetry and C is transitivity. If all goes as planned, the result is the cramming of proof steps of symmetry into a proof of transitivity. By the way, I use cramming heavily, both in the case in which the target is the join or simultaneous consideration of two or more targets and in the case in which a single formula or equation is the target. Just for clarity, in the latter case, my goal is to force certain deduced steps to be used repeatedly, as often as possible, and thus avoid the need for so-called extra steps whose presence might result in the completion of a proof longer than desired.

Of course, as you might have surmised, I also run different experiments, where the difference resides with various values assigned to max_weight and other such parameters. Common to the various approaches is the intent to perturb the wanderings through the huge search space of deducible conclusions. As you may know or will discover with experimentation, a large obstacle is presented by the hugeness of the space of conclusions that can be deduced. Even with a very fast computer and a very fast program, you will find it far more than advisable to rely on restriction strategies and direction strategies, rather than merely having the program wander wherever it happens to go. For but one example, most pertinent to the expeditions narrated here, almost for certain, all proofs showing XCB to be a single axiom, terminating with the deduction of the cited 2-basis or with the deduction of a known single axiom, require the presence of at least one formula relying on twelve distinct variables. In the context of symbol count, (including the predicate symbol) max_weight must be assigned a value of at least 48 if a proof is to be found, and any exhaustive search within such a constraint is doomed.

Now you know a fair amount about the mode of travel, about the so-called equipment that was used, about the nature of the experiments. If you and I were reading this story or notebook for the first time, you might share the impatience I would certainly be feeling. Indeed, precisely what treasure was found? Certainly, at least some of you would prefer to try your own hand at mining, rather than being treated to a huge number of details regarding one experiment after another. Further, in that I run numerous experiments simultaneously, I suspect I would have difficulty recovering the actual journey as it occurred.

If memory serves, Ulrich was most interested in seeking a proof completing with a deduction of the two-element basis, that consisting of symmetry and transitivity. I still know of no way to capture with OTTER the aforementioned skill that Ulrich has, of studying a proof and discovering how a step can be saved. But I was able to rely on the cited approaches. In a bit more than two weeks after the astounding and most satisfying success with proving XCB a single axiom, on May 1, 2002, OTTER presented a 25-step proof of the 2-basis. However, history should note that OTTER first found a 26-step proof. It was able to complete the cited 25-step proof because Ulrich showed me how to save a step. Especially for the curious, Ulrich then or soon after set forth the goal of finding a proof of length strictly less than 20, a goal still out of reach. Nevertheless, he and I were both delighted with the proof we had. By the way, Ulrich and I each conjectured that, to find a proof of length less than 25, formulas of greater complexity than present in our proof would probably be needed. The 25-step proof requires the use of formulas of weight (complexity) 48 (measured in symbol count), including predicate, formulas relying on twelve distinct variables. Further, the experiments suggest that no proof showing XCB to be a single axiom and completing with the deduction of a known axiom system can be found with strictly less complexity.

By early July 2002, more treasure had been extracted. In particular, OTTER had presented proofs of lengths between 26 and 30 of the other thirteen shortest single axioms. In addition, the program had found seven other proofs of known single axioms, each axiom of length greater than eleven, whose lengths range from 29 to 36.

If I recall correctly, Ulrich and I were each quite pleased at the reduction in proof length, especially in the context of the 2-basis. Further experiments yielded but a few small nuggets. Of course, as some who know me would assert, in the back of my mind was the intention of revisiting this area when and if new and powerful strategies were forthcoming. But I was not very optimistic; after all, many, many additional experiments had proved essentially unsuccessful.

The Second Expedition

In mid-summer 2004, Argonne hosted one of a series of workshops on automated reasoning and deduction. Present at that workshop was Mark Stickel, a scholar indeed. At one of my talks, I mentioned the goal of finding a proof of length strictly less than twenty-five, showing XCB to be a single axiom. I did not expect anybody to seriously consider the problem at that time. However, months later, on December 24, 2004, I received an exciting e-mail from Mark--what a Christmas present!

Mark had access to a program, of his own design, that thoroughly explored the use of condensed detachment, given an axiom system. As for the cited excitement, Mark sent six proofs that deduced, from XCB, the 2-basis of interest, each proof of length 24. Yes, he had found proofs of length less than 25, a goal that had resisted my substantial attempts. The first three proofs were of greater interest in that the last three of the six were variations of the first three, differing in their history. Now, as may have become clear at this point, the history is not my main concern; rather, the set of formulas that are deduced that together (in effect) form a proof are the items on which I focus. What a treat: to have in hand a new and shorter proof to study and experiment with!

I informed Ulrich of Mark's most satisfying achievement, and, of course, he was pleased and interested. What was Mark's secret? What insight did he have that had eluded me? Well, harking back to months and months earlier (as remarked earlier), Mark had permitted his program to consider formulas relying on strictly more than 48 symbols. Indeed, in his first proof and his third proof, one of the formulas relies on sixteen distinct variables that, if measured in symbol count (including predicate), has weight 64. Now what do you think I did with Mark's proof, and, of not quite as much interest, which proof did I focus on?

The second question is easier to answer, especially for those who know of my lack of patience--some call it haste. I chose the first proof I saw among the six, totally paying no attention to the other five. As for the first question, I (as one might predict) used the twenty-four formulas of Mark's proof as hints (and no others, if memory serves) and employed McCune's ancestor subsumption. In less then 2 CPU-minutes, still on December 24, OTTER offered a newer and most pleasing proof, a proof of length 23. I note immediately, for any person who concludes that Mark had missed something, that ancestor subsumption is quite powerful. Further, the deduction of the 2-basis relied on two clauses, each of which is numbered in excess of 55,000 (marking how many clauses had been retained, and who knows how many had been deduced). Mark certainly did not miss anything; instead, he merits substantial compliments for his achievement.

I was, of course, presented with a problem. Should I notify Mark, which might in some way bring him displeasure, or should I simply sit on the 23-step proof, which I had sent to Ulrich as soon as I had it? The decision was to send the 23-step proof to Mark, who was most gracious.

Nobody (familiar with my attitude toward research) can doubt my zeal to find a proof of length strictly less than 23 (applications of condensed detachment). After all, besides the greed I always experience, Ulrich had told me of his goal to find a proof of length strictly less than 20, not requiring the 2-basis as the termination point. So I tried, and tried, and tried, but to no avail. Indeed, if I recall correctly--as I sit here, I'm trying to talk about occurrences of late 2004 and early 2005, and now it is July 2007--I renewed this treasure hunt on and off for many weeks.

In early April 2005 (April 2, I am fairly certain), I finally turned to the use of the cramming strategy. Specifically, I added in the spirit of axioms the first seven steps of the 23-step proof. To force the program to use each of the seven to initiate applications of condensed detachment, I placed them in list(sos) and added the command list(sos_queue) to instruct OTTER to use level saturation. After running overnight and retaining more than 28,000 new clauses, OTTER informed me that fifteen additional deduced steps would complete a proof of the 2-basis. Yes, if taken together, the first seven steps of the 23-step proof combined with the appropriately chosen fifteen steps would produce a 22-step proof, the first of its kind. Indeed, a run with those twenty-two hints immediately verified the claim, and (for Ulrich and for me) more fine treasure had been found. For the person enjoying piquant details, the 22-step proof relies on seven formulas not present in the 23-step proof--so I claim, if my delving back into history has been successful.

As for the other thirteen single axioms, except for P9 and XHN, this second expedition found shorter proofs than were found in the first expedition. Especially satisfying was a proof of length 24 deducing P7. Of the other seven targets, the single axioms of length greater than 11, only that called Wajsberg_5 had its proof shortened. And, at this point, I might have thought the seeking of shorter proofs of the kind in focus here was at an end--except, knowing myself as I do, such a search is essentially never at an end. Indeed, when and if a new strategy or idea should be encountered, I knew that I would again revisit equivalential calculus in the context of XCB. And I did.

The Third Expedition

This third expedition was prompted by Overbeek, suggesting that I offer various notebooks concerning aspects of my past, present, and, perhaps most important, possible future research. I thought that, since I was going to attempt to recapture some of history, I should again seek shorter proofs by relying on something. It occurred to me that I should revisit Mark's e-mail of December 2004, inspecting his proofs a bit closer than I had. The second proof, after the smallest number of experiments, was discarded in the context of seeking shorter proofs, mainly because it did not rely on enough distinct variables, and I thought additional variable richness might be needed. However, the third proof appeared promising, especially in that it relied on 16-variable formulas as the first does. And the third expedition was under way.

Since the approach taken in the second expedition had yielded new results, I decided (not profoundly) to emulate it, taking his third (24-step) proof and seeking a shorter proof from it. I relied on the twenty-four formulas from Mark's proof as hints, and I used ancestor subsumption. In less than 90 CPU-seconds, OTTER returned to me a 23-step proof. This new proof differed from the preceding 23-step proof (that played such a key role) only in one important respect: Its fifth step. In fact, except for the cited difference, all of the formulas occur in the same order in both proofs. Could such a small difference ever lead to any new discoveries?

Fortunately, with OTTER as an assistant, virtually no effort was required to find out. As before, I used cramming, placing the first seven steps of the new 23-step proof in list(sos), and placing in the hints list various formulas from earlier proofs. As noted, the hints are included to give some guidance to the program's search. As in the earlier study with the earlier 23-step proof, I instructed OTTER to rely on level saturation, hoping that a new proof would be found, one of interest. After a long, long run, nearly 11,000 CPU-seconds, and the retention of almost 26,000 new conclusions, OTTER found sixteen steps that, taken together with the given seven, yielded a proof of the 2-basis.

Rather than assuming no progress had been made, I included the sixteen formulas as hints, still included the cited seven, as well as hints from an earlier set of proofs. The goal (hope might be better advised) was that the program would not merely present the already-in-hand 23-step proof of the 2-basis. It did not; instead, it almost immediately offered a 22-step proof, one that, with a bit of analysis was slightly different from the 22-step proof already found. In particular, the eighteenth step of the older 22-step proof was replaced by a different formula; the former eighteenth step relies on eleven distinct variables, and the latter on eight. In other words, the newer 22-step proof has smaller size—1658 versus 1696—a term (introduced to me by Ulrich) that measures the total number of symbols present in the deduced formulas. The new 22-step proof contains five formulas not in the newer 23-step proof. So, although not a proof shorter in length, some progress had occurred.

More was to be found in the corresponding output file; indeed, as is typical of my research, I instructed the program to accrue as many proofs as it could by running for a long, long time. Specifically, for quite a few of the other thirteen shortest single axioms, a shorter proof was found. Most exciting to me was the finding, among the new proofs, two of length 23, that for P2_YQF and that for P7_WN, the first proofs of that length terminating in the deduction of a known single axiom. I had never before seen proofs this short that deduced from XCB another shortest single axiom. Yes, I still consider the cited results a treasure.

The expedition was nearing its end. Of course many more experiments occurred to me. One of the obstacles rests with the huge number of experiments that can be tried. I did not simply return to my home with the cited treasure in hand. I wished to find even shorter proofs for all of the twenty targets, the thirteen shortest single axioms (other than XCB) as well as the seven of length greater than eleven. What had I not tried? After all, continued cramming and modifying of the value of various parameters was yielding nothing of significance. And an idea occurred to me, one that I had not used (as far as I know) in any of the three expeditions. I could add to the approach the use of the hot list strategy, which I shall discuss briefly right now.

The hot list strategy came into being because of one general observation I made, one that would be evident to anyone reading a number of proofs in mathematics and in logic. In particular, quite common to a proof in either field is frequent use (or return to) the key property or properties in focus. For example, if the study concerns rings in which the cube of every element x is x, and if the goal is to prove that such rings are commutative, a typical proof will rest on frequent use of the added property, that concerning the cube of x. To have a program offer this type of approach, the hot list strategy was born. To use it with, say, OTTER, you place in a new list, list(hot), the item or items you wish to take center stage. To apply the strategy, the program must then, upon the retention of a new conclusion, apply the inference rule or rules that are being used to the newly retained conclusion together with the various elements of the hot list.

In the case of a study of XCB, you place in list(hot) a copy of the corresponding clause as well as a copy of the clause for condensed detachment. You also include a command of the form assign(heat,1). These actions will cause the program to visit XCB (with condensed detachment) every time a new conclusion is retained.

Clearly, my greatest objective was to find a proof of length 21 or less. Failing that, I was after shorter proofs of some of the twenty targets. I can report partial success: Some new and shorter proofs were found, which caused me to consider closing this notebook and ending the third Expedition. After all, with few exceptions, at this point, the various proofs were of length strictly less than thirty.

Nevertheless, one final experiment occurred to me. The object was to remove at least one of the exceptions--to find a proof, for one of those annoying formulas, of length strictly less than thirty. I placed in the hints list only correspondents of a 30-step proof of Wajsberg_4 and those of a 31-step proof of Bryman. Perhaps OTTER would return to me one of the desired proofs, of length twenty-nine or less. I was, as you would note, trying to build on earlier successes with proof shortening. Research sometimes works that way: You make one or more discoveries, and then you use those results to obtain even more. To my satisfaction, progress did occur. To my amazement, progress was made regarding various axioms, namely, P1_YQL, P9_YRO, Bryman, one of the Lukasiewicz axioms, and both Wajsberg axioms. Why the use of the cited hints had such a profound effect--especially after I had conducted such a large number of experiments--is beyond my current understanding. If you wish to tackle a substantial challenge, you might try to provide a good explanation, or at least a more-than-plausible conjecture.

Perhaps one of you who reads this will find even shorter proofs. If you are considering an attempt, I now offer you a possibly useful table. Also, because this writing is mostly like a notebook, I offer you a brief treatment of yet one more experiment, one of a nature different from those discussed earlier, and one that can be thought of as an interjection or a P.S.

I had believed that the notebook was as complete as it was going to be. It evinced, before the experiment I now discuss, one small drawback, namely, all proofs but one that completed with the deduction of a shortest single axiom had lengths between 23 and 26. The formula XGK was the exception; indeed, after numerous tries, I had not been able to find a proof of length strictly less than 27. So, it seemed, I would have to live with the exception. Then it occurred to me that the blocking of formulas with demodulation (demodulating the unwanted to junk) also blocked instances of an unwanted formula, but an alternative existed. In particular, by including an item of the following type, the program will, instead of blocking instances, block similar formulas, those differing only at the variable level (treating all variables as indistinguishable).

weight(P(e(e(e(e(e(x,y),e(z,y)),z),u),e(x,u))),100).

With this inclusion and a max_weight assigned a value strictly less than 100, when and if the given formula or one similar to it is deduced, it will be discarded. I chose that formula because it was used only once as a parent in a 27-step, level-19 proof of XGK. Yes, more treasure: In less than 1 CPU-second, OTTER returned a 26-step, level-22 proof, and the table I now give was to me more pleasing, all proofs terminating in the deduction of a shortest single axiom have lengths between 23 and 26.

Table of Proof Lengths

In the following table, you will find the treasures found in the respective expeditions. The results are listed in the following order: those from the third Expedition, from the second, from the first, and from the original run that first proved XCB to be a single axiom. The table designates which expedition is in focus and which target is in focus, the 2-basis followed by known single axioms. If you decide to study any of the individual targets with XCB as the sole hypothesis, you will have in hand the best I was able to find as of mid-2007.

#3 2-basis 22, 724 size not counting pred parens commas;
#2 22, size 744; #1 22; #0 71;
#3 P1 24; #2 26; #1 28; #0 76
#3 P2 23; #2 25; #1 26; #0 78;
#3 P3 26; #2 27; #1 29; #0 77;
#3 P4 26; #2 27; #1 29; #0 75;
#3 P5 25; #2 27; #1 29; #0 77;
#3 P7 23; #2 24; #1 26; #0 76;
#3 P8 24; #2 26; #1 27; #0 77;
#3 P9 26; #2 29; #1 29; #0 75;
#3 PYM 25; #2 26; #1 29; #0 77;
#3 PYO 25; #2 27; #1 28; #0 87;
#3 XGK 26; #2 27; #1 30; #0 81;
#3 XHK 26; #2 26; #1 29; #0 77;
#3 XHN 26; #2 29; #1 29; #0 77;
#3 Bry 29; #2 34; #1 34;
#3 Luka1 27; #2 33; #1 33;
#3 Luka2 28; #2 36; #1 36;
#3 Sobo1 28; #2 29; #1 29;

#3 Sobo2 26; #2 28; #1 28;
#3 Wajs4 29; #2 36; #1 36;
#3 Wajs5 27; #2 31; #1 35; #0 79;

Epilogue

I intend that this be but one of a series of notebooks or articles on the automation of reasoning. Although each is clearly not polished, nevertheless a reading of one or more of them might provide you with some excitement. Further, perhaps you will find a thesis topic in one of these notebooks or the basis for research that leads to a publication.

At this point, as promised, I focus on some history provided to me by Ulrich, and, in that context, I offer you a challenge. The very first SINGLE axioms for EC ever found are of length 15 and are due to Wajsberg. Six additional single axioms of length 15 were discovered next, one by Bryson, two by Lukasiewicz, and three by Sobocinski. The SHORTEST POSSIBLE single axioms for EC are of length 11. The first three such axioms were found by Lukasiewicz (YQL, YQF, and YQJ), the next 7 by C. A. Meredith (UM, XGF, WN, YRM, YRO, PYO, and PYM), the eleventh by Kalman/Meredith (XGK); note that Kalman himself regards it as the mere correction of a typographical error of an eighth one of Meredith's. The twelfth and thirteenth were found by S. Winker (XHK and XHN), and the fourteenth by us (XCB). Ulrich included a third axiom by Sobocinski, one that I had never studied, which I offer to you in two contexts. You can take it and try to derive some known basis, and, more in the spirit of this notebook, you can try to find a short proof showing that XCB implies it, the following.

P(e(e(x,e(y,z)),e(e(x,e(z,u)),e(y,u)))). % Sobocinski 3

The notion of automating logical reasoning is in one obvious sense preposterous, if the goal is to provide aid for serious and dedicated scientists. However, that is exactly what has happened. Indeed, various open questions in mathematics and logic have been answered, some that were open for many decades. My goal is not only to cause more activity in science. In fact, if one of you becomes in some form addicted, you will share in the delight that I more than occasionally experience. I find it more than stimulating when OTTER, at my direction and restriction, completes a proof that is in various ways better than the literature offers. If you improve on one of the results cited here, or certainly better, on some result that has been published, I am quite sure you will experience exhilaration. If you can find a copy, and if you have enjoyed some of what you have read here, you might enjoy a book I wrote some years ago, The Automation of Reasoning: An Experimenter's Notebook with OTTER Tutorial, L. Wos, Academic Press, New York, 1996.

I promised to include an input file to show you how you might begin. Here it is.

Input File

```
set(hyper_res). assign(max_mem,680000). % assign(max_seconds,10). % set(sos_queue).
assign(max_weight,72). assign(change_limit_after,1100). assign(new_max_weight,10).
assign(max_proofs,-1). assign(pick_given_ratio,4). assign(bsub_hint_wt,1).
clear(keep_hint_subsumers). set(keep_hint_equivalents). set(ancestor_subsume). set(back_sub).
set(order_history). % set(process_input).
clear(print_kept).

list(usable).
-P(e(x,y)) | -P(x) | P(y).
-P(e(e(a,b),e(b,a))) | -P(e(e(a,b),e(e(b,c),e(a,c)))) | $ANSWER(all_s_t_indep).
end_of_list.

list(sos).
```

```
 P(e(x,e(e(e(x,y),e(z,y)),z))).  % XCB
 end_of_list.

 list(demodulators).  %  (P(e(e(e(e(e(e(x,e(e(e(x,y),e(z,y)),z)),u),v),e(u,v)),e(e(w,v6),e(v7,v6)),v7)),w)) =
junk).  %  (e(e(x,x),y) = junk).  %  (e(y,e(x,x)) = junk).
 (e(x,junk) = junk).
 (e(junk,x) = junk).
 (P(junk) = $T).
 end_of_list.

 list(passive).  %  Following are axioms for EC and other targets.
 -P(e(e(a,b),e(e(c,b),e(a,c)))) | $ANSWER(P1_YQL).
 -P(e(e(a,b),e(e(a,c),e(c,b)))) | $ANSWER(P2_YQF).
 -P(e(e(a,b),e(e(c,a),e(b,c)))) | $ANSWER(P3_YQJ).
 -P(e(e(e(a,b),c),e(b,e(c,a)))) | $ANSWER(P4_UM).
 -P(e(a,e(e(b,e(a,c)),e(c,b)))) | $ANSWER(P5_XGF).
 -P(e(e(a,e(b,c)),e(c,e(a,b)))) | $ANSWER(P7_WN).
 -P(e(e(a,b),e(c,e(e(b,c),a)))) | $ANSWER(P8_YRM).
 -P(e(e(a,b),e(c,e(e(c,b),a)))) | $ANSWER(P9_YRO).
 -P(e(e(e(a,e(b,c)),c),e(b,a))) | $ANSWER(PYO).
 -P(e(e(e(a,e(b,c)),b),e(c,a))) | $ANSWER(PYM).
 -P(e(a,e(e(b,e(c,a)),e(c,b)))) | $ANSWER(XGK).
 -P(e(a,e(e(b,c),e(e(a,c),b)))) | $ANSWER(XHK).
 -P(e(a,e(e(b,c),e(e(c,a),b)))) | $ANSWER(XHN).
 -P(e(a,a)) | $ANSWER(reflex).
 -P(e(e(a,b),e(b,a))) | $ANSWER(symm).
 -P(e(e(a,b),e(e(b,c),e(a,c)))) | $ANSWER(trans).
 -P(e(e(e(c1,e(c2,c3)),e(e(c3,c4),c4)),e(c1,c2))) | $ANSWER(Wajsberg_4_sing).
 -P(e(e(e(e(c1,c2),c3),c4),e(c4,e(c1,e(c2,c3))))) | $ANSWER(Wajsberg_5_sing).
 -P(e(e(c1,e(c2,c3)),e(e(c2,e(c4,c3)),e(c4,c1)))) | $ANSWER(Bryman_sing).
 -P(e(e(c1,e(c2,c3)),e(e(c2,e(c3,c4)),e(c4,c1)))) | $ANSWER(Luka_1_sing).
 -P(e(e(c4,e(c1,e(c2,c3))),e(e(c1,c2),e(c3,c4)))) | $ANSWER(Luka_2_sing).
 -P(e(e(c1,e(c2,c3)),e(e(c1,e(c3,c4)),e(c4,c2)))) | $ANSWER(Sobo_1_sing).
 -P(e(e(c1,e(c2,c3)),e(e(c1,e(c4,c3)),e(c4,c2)))) | $ANSWER(Sobo_2_sing).
 end_of_list.

 list(hints).  %  Following first 7 of a new 23-step proof of the 2-basis, %  from XCB, 07-05-07, a proof
whose fifth step is %  different from the fifth of the much %  earlier 23; the new 23 found by using Stickel's
3rd proof, %  rather than his first.
 P(e(e(e(e(x,e(e(e(x,y),e(z,y)),z)),u),e(v,u)),v)).
 P(e(e(e(e(x,e(e(e(x,y),e(z,y)),z)),u),v),e(u,v))).
 P(e(e(e(e(e(e(e(x,e(e(e(x,y),e(z,y)),z)),u),v),e(u,v)),w),e(v6,w)),v6)).
 P(e(x,e(e(e(e(e(y,e(e(e(y,z),e(u,z)),u)),x),v),e(w,v)),w))).
 P(e(x,e(e(e(e(e(y,e(e(e(y,z),e(u,z)),u)),e(e(v,e(e(e(v,w),e(v6,w)),v6)),x)),v7),e(v8,v7)),v8))).
 P(e(x,e(e(e(e(e(y,e(e(e(y,z),e(u,z)),u)),e(e(v,e(e(e(v,w),e(v6,w)),v6)),e(e(v7,e(e(e(v7,v8),e(v9,v8)),v9)),x))),v10),e(v11,v10)),v11))).
 P(e(e(e(e(x,e(e(e(x,y),e(z,y)),z)),e(e(u,e(e(e(u,v),e(w,v)),w)),e(e(v6,e(e(e(v6,v7),e(v8,v7)),
v8)),e(e(e(e(e(v9,e(e(e(v9,v10),e(v11,v10)),v11)),v12),v13),e(v12,v13)),v14))))),v15),e(v14,v15))).  %  Fol-
lowing 15 from cramming on preceding 7, in temp.xcb.new.out1f.
 P(e(e(e(e(e(x,e(e(e(x,y),e(z,y)),z)),e(u,e(e(e(u,v),e(w,v)),w))),v6),e(v7,v6)),v7)).
 P(e(e(e(e(e(x,e(e(e(x,y),e(z,y)),z)),e(u,e(e(e(e(e(v,e(e(e(v,w),e(v6,w)),v6)),e(e(v7,
e(e(e(v7,v8),e(v9,v8)),v9)),u)),v10),e(v11,v10)),v11))),v12),e(v13,v12)),v13)).
 P(e(e(x,e(e(y,e(e(e(y,z),e(u,z)),u)),x)),e(v,e(e(e(v,w),e(v6,w)),v6)))).
 P(e(e(x,e(e(y,e(e(e(e(e(z,e(e(e(z,u),e(v,u)),v)),e(e(w,e(e(e(w,v6),e(v7,v6)),
```

v7)),y)),v8),e(v9,v8)),v9)),x)),e(v10,e(e(e(v10,v11),e(v12,v11)),v12)))).
P(e(e(e(e(e(x,e(e(y,e(e(e(y,z),e(u,z)),u)),x)),e(v,e(e(e(v,w),e(v6,w)),v6))),v7),e(v8,v7)),v8)).
P(e(e(e(x,e(y,e(e(e(y,z),e(u,z)),u))),v),e(x,v))).
P(e(e(e(x,y),x),y)).
P(e(e(e(x,e(e(e(x,y),e(z,y)),z)),e(e(e(u,v),e(w,v)),w)),u)).
P(e(e(e(e(x,y),e(z,y)),z),x)).
P(e(e(e(e(e(e(e(x,y),e(z,y)),z),x),u),e(v,u)),v)).
P(e(e(e(e(x,e(e(e(x,y),e(z,y)),z)),e(e(e(e(u,v),e(w,v)),w),u),v6)),v7),e(v6,v7))).
P(e(e(x,y),e(e(y,z),e(x,z)))).
P(e(e(x,y),e(e(e(z,x),z),y))).
P(e(e(e(e(e(x,y),x),z),u),e(e(y,z),u))).
P(e(e(x,y),e(y,x))).
end_of_list.

I also promised to include a proof; I have chosen the latest proof, and smallest in size, of the 2-basis.

A Fine Proof

----- Otter 3.3g-work, Jan 2005 -----
The process was started by wos on lemma.mcs.anl.gov,
Mon Jul 23 10:12:00 2007
The command was "otter". The process ID is 27347.

-----> EMPTY CLAUSE at 0.06 sec ----> 423 [hyper,2,403,274] $ANSWER(all_s_t_indep).

Length of proof is 22. Level of proof is 17.

---------------- PROOF ----------------

1 [] -P(e(x,y)) | -P(x) | P(y).
2 [] -P(e(e(a,b),e(b,a))) | -P(e(e(a,b),e(e(b,c),e(a,c)))) | $ANSWER(all_s_t_indep).
3 [] P(e(x,e(e(e(x,y),e(z,y)),z))).
52 [hyper,1,3,3] P(e(e(e(e(x,e(e(e(x,y),e(z,y)),z)),u),e(v,u)),v)).
54 [hyper,1,52,3] P(e(e(e(e(x,e(e(e(x,y),e(z,y)),z)),u),v),e(u,v))).
56 [hyper,1,3,54] P(e(e(e(e(e(e(e(x,e(e(e(x,y),e(z,y)),z)),u),v),e(u,v)),w),e(v6,w)),v6)).
58 [hyper,1,54,3] P(e(x,e(e(e(e(e(y,e(e(e(y,z),e(u,z)),u)),x),v),e(w,v)),w))).
64 [hyper,1,54,58]
P(e(x,e(e(e(e(e(y,e(e(e(y,z),e(u,z)),u)),e(e(v,e(e(e(v,w),e(v6,w)),v6)),x)),v7),e(v8,v7)),v8))).
70 [hyper,1,58,3] P(e(e(e(e(e(x,e(e(e(x,y),e(z,y)),z)),e(u,e(e(e(u,v),e(w,v)),w))),v6),e(v7,v6)),v7)).
77 [hyper,1,58,64]
P(e(e(e(e(e(x,e(e(e(x,y),e(z,y)),z)),e(u,e(e(e(e(e(v,e(e(e(v,w),e(v6,w)),v6)),e(e(v7,e(e(e(v7,v8),e(v9,v8)),v9)),u)),v10),e(v11,v10)),v11)
79 [hyper,1,54,64]
P(e(x,e(e(e(e(e(y,e(e(e(y,z),e(u,z)),u)),e(e(v,e(e(e(v,w),e(v6,w)),v6)),e(e(v7,e(e(e(v7,v8),e(v9,v8)),v9)),x))),v10),e(v11,v10)),v11))).
102 [hyper,1,56,79]
P(e(e(e(e(x,e(e(e(x,y),e(z,y)),z)),e(e(u,e(e(e(u,v),e(w,v)),w)),e(e(v6,e(e(e(v6,v7),e(v8,v7)),v8)),e(e(e(e(v9,e(e(e(v9,v10),e(v11,v10)),
132 [hyper,1,102,77]
P(e(e(x,e(e(y,e(e(e(e(e(z,e(e(e(z,u),e(v,u)),v)),e(e(w,e(e(e(w,v6),e(v7,v6)),v7)),y)),v8),e(v9,v8)),v9)),x)),e(v10,e(e(e(v10,v11),e(v12,v1
133 [hyper,1,102,70] P(e(e(x,e(e(y,e(e(e(y,z),e(u,z)),u)),x)),e(v,e(e(e(v,w),e(v6,w)),v6)))).
148 [hyper,1,3,133]
P(e(e(e(e(e(x,e(e(y,e(e(e(y,z),e(u,z)),u)),x)),e(v,e(e(e(v,w),e(v6,w)),v6))),v7),e(v8,v7)),v8)).
154 [hyper,1,148,133] P(e(e(e(x,y),x),y)).
155 [hyper,1,148,132] P(e(e(e(x,e(y,e(e(e(y,z),e(u,z)),u))),v),e(x,v))).
212 [hyper,1,155,52] P(e(e(e(x,e(e(e(x,y),e(z,y)),z)),e(e(e(u,v),e(w,v)),w)),u)).

220 [hyper,1,54,212] P(e(e(e(e(x,y),e(z,y)),z),x)).
230 [hyper,1,3,220] P(e(e(e(e(e(e(x,y),e(z,y)),z),x),u),e(v,u)),v)).
263 [hyper,1,230,58] P(e(e(e(e(x,e(e(e(x,y),e(z,y)),z)),e(e(e(e(u,v),e(w,v)),w),u),v6)),v7),e(v6,v7))).
274 [hyper,1,263,220] P(e(e(x,y),e(e(y,z),e(x,z)))).
305 [hyper,1,274,154] P(e(e(x,y),e(e(e(z,x),z),y))).
323 [hyper,1,274,305] P(e(e(e(e(e(x,y),x),z),u),e(e(y,z),u))).
403 [hyper,1,323,220] P(e(e(x,y),e(y,x))).
423 [hyper,2,403,274] $ANSWER(all_s_t_indep).

------------ end of proof -------------

Perhaps surprising is the fact that, more than the actual finding of treasure, the hunt for many is where the real excitement resides. For me, nevertheless, the finding was equally exciting as the hunt was.

As for an application of proof shortening, I conjecture that certain textbooks could be made simpler by reducing the length of various proofs to the point that certain thought-to-be key lemmas would in fact be avoided. Such lemma avoidance has occurred in my research.

As for ideas and challenges for the future, the following occur to me. You might take three steps in the middle of a proof, three thought to be crucial. Take the negation of their join and seek a short proof of such, which will force the proofs of each of the three to share steps. If you are successful, the short subproof might be useful in leading to a shorter proof of the original goal.

As for additional challenges, you might consider seeking a proof whose length is strictly less than that cited in the table. If you enjoy programming, you might attempt to automate the type of expedition in focus here.

Until we meet again in another notebook, I await any communication you wish to make. My e-mail address is wos@mcs.anl.gov . I do plan to produce additional notebooks of this type.

Acknowledgments

I thank D. (Ted) Ulrich greatly for sharing with me various studies of equivalential calculus. My discussions with him were most exciting and extremely informative. His influence on my research will be found in other notebooks. I thank Mark Stickel for sharing his most satisfying 24-step proof for XCB. His research certainly influenced mine substantially. I thank Ross Overbeek and Michael Beeson for their encouragement. Each helped me to decide to place this material, and similar works, on the Web. I thank Gail Pieper for her tireless reading and most useful suggestions. I thank various members of the Mathematics and Computer Science Division of Argonne National Laboratory for continued, unfailing support. This work was supported in part by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, Office of Science, U.S. Dept. of Energy, under Contract DE-AC02-06CH11357.