ri
created 11-02-2007
revised last 06-12-2009

A Startling Result, Some Challenges Met, but Some Still Remain: More Coping with Complex Expressions*

Larry Wos

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

1. Brief Introduction

In this notebook, you will learn of a most unusual collaboration, one that (in effect) involves three people. The story begins with another marvelous success of the logician D. (Ted) Ulrich, person number one. That success caused me, person number two, to attempt to refine his result. You will learn that such did occur, but, far more interesting to many, you will be told of a type of problem that, ordinarily, is quite difficult, or virtually impossible, to consider for a program like William McCune's OTTER (featured here). (As in the various notebooks found on this website, the approaches that are offered d not require the use of OTTER; see http://www.mcs.anl.gov/AR/otter/.) The fault, dear readers, does not lie with OTTER; rather, the fault rests with the complexity of certain formulas. Enter person number three, Ross Overbeek, whose ideas of decades ago play a huge role in the story to unfold. The confluence of the contributions of these three people led to a new approach to proof finding, when the target is a formula of substantial complexity. But let us first focus on the innocent inception of what is to follow.

Ulrich studies many areas of logic, placing many of his charming successes on his website, http://web.ics.purdue.edu/~dulrich/. He and his program (which he has relied upon for many, many years) often seek single axioms for some area of logic, for example (as discussed in an earlier notebook), an area known as *BCK* and an area known as *BCI*. Sometimes—and most pertinent to this notebook—Ulrich finds single axioms that are shorter than previously offered by the literature.

Quite recently, he turned to the study of an area called *RI*, which is the implicational fragment of *R*. The area *RI* can be axiomatized with the following four formulas, expressed as clauses, where the function *i* denotes implication, the predicate *P* denotes "provability", and variables (such as *u*, *V*, *W*, *x*, *V*6) are universally quantified (meaning "for all").

P(i(i(u,v),i(i(v,w),i(u,w)))). % B'
P(i(i(u,i(v,w)),i(v,i(u,w)))). % C
P(i(u,u)). % I
P(i(i(u,i(u,v)),i(u,v))). % W

You can with no real penalty replace *B*' by the following formula, *B*.

P(i(i(x,y),i(i(z,x),i(z,y)))). % B

(Ulrich and I often prefer *B*' to *B*, perhaps because it reminds one of transitivity.)

As for the area *RI* before Ulrich studied it, there in fact did exist a single axiom, one due to the Romanian logician Adrian Rezus. He has a way of producing single axioms for various areas of logic

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

provided that certain conditions are satisfied. His single axiom, the following, has length 93, measured in symbol count as a formula (excluding the predicate symbol, commas, and parentheses).

```
% Following is a Rezus-style single axiom for RI.
P(i(i(i(x,i(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),w)),i(i(i(i(v6,v7),i(i(v7,v8),i(v6,v8))),
  i(i(i(i(v9,i(v9,v10)),i(v9,v10)),i(i(i(v11,v12),i(i(v12,v13),i(v11,v13))),v14)),v14),v15)),v15),
  (i(v16,i(i(i(v17,v17),i(i(v18,v18),i(i(v19,v19),i(i(v20,v20),i(v16,v21))))),v21)),v22))),v22)).
% Rezus-style
```

Ulrich set himself the task of finding a shorter, perhaps far shorter, single axiom. You might immediately wonder about earlier such studies, studies aimed at finding single axioms and, more important, finding a single axiom when none known at the time. Well, classical propositional calculus admits a number of axiom systems including one due to Hilbert. Lukasiewicz found in the 1930s the following 23-letter single axiom.

```
% Following is Lukasiewicz's 23-letter single axiom.
P(i(i(i(x,y),i(i(i(n(z),n(u)),v),z)),i(w,i(i(z,x),i(u,x))))).
```

Many years later, the logician C. A. Meredith improved on the Lukasiewicz result, offering the following 21-letter single axiom for classical propositional calculus.

```
% Following is Meredith's axiom.
P(i(i(i(i(i(x,y),i(n(z),n(u))),z),v),i(i(v,x),i(u,x)))).
```

You thus have one example (among many) that illustrates the interest shown by fine minds in the pursuit of shorter single axioms.

A glance at Ulrich's website correctly suggests that he is indeed successful in such a pursuit, and in diverse areas of logic. In some cases, his results are astounding. Indeed, in the context of *RI*, he offered the world the following 35-symbol single axiom (expressed as a clause).

```
% Following is Ulrich's axiom.
P(i(i(i(i(i(u,v),i(w,u)),i(i(u,v),i(w,v))),i(i(x,x),i(y,i(z,v6)))),i(i(v7,y),i(z,i(v7,v6))))).
```

When he conveyed his success to me, I naturally turned to the aspect of research that clearly intrigues me, namely, the pursuit of short proofs. As will be learned here, my interest in the Ulrich axiom led to a significant breakthrough, in the context of targets that are quite complex, a breakthrough that was sought for decades.

That bit of research is in part the focus in this notebook. Also in focus, later, is my study of a Rezus-style single axiom, a study that provided a significant test for new strategy (to be discussed), namely, the *subformula strategy*. That new strategy was born, not with the study of the Rezus-style axiom, but with the prior study of the Ulrich 35-symbol single axiom.

2. Seeking Short Proofs

Two studies are asked for. In one, the object is to prove that from, say, the Ulrich 35-symbol formula, you can deduce a known axiom system. In the other, to show the (in this case) the Ulrich formula is a theorem, you are asked to deduce it from a known axiom system. Of course, my goals were to find proofs in each study but, as expected, to find "short" proofs. I first turned to using the Ulrich formula as the sole hypothesis.

So typical of much of my research concerning finding short proofs, I began the attempt with a proof supplied by Ulrich. That proof, consisting of twenty steps of condensed detachment, deduces from his 35-symbol formula *B*', *C*, *I*, and *W*. My first try at finding a shorter proof (than length 20) was to submit the following input file in which I used R. Veroff's *hints strategy* and W. McCune's ancestor subsumption. (I briefly discuss each after giving the following input file.)

An Input File For Seeking a Proof of Length Less Than 20

```
set(hyper_res).
assign(max_mem,880000).
% 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,1).
assign(bsub_hint_wt,1).
set(keep_hint_subsumers).
set(ancestor_subsume).
set(back_sub).
set(order_history).
% set(process_input).
clear(print_kept).

list(hints).
% Following 20 derive B' C I and W from the new Ulrich 35-symbol single axiom for RI.
P(i(i(u,v),i(i(v,v),i(u,v)))).
P(i(i(u,i(i(v,w),i(x,v))),i(i(v,w),i(u,i(x,w))))).
P(i(i(u,i(i(v,v),i(w,v))),i(w,i(u,v)))).
P(i(i(u,i(i(i(v,w),i(v,w)),i(i(x,x),i(v,w)))),i(v,i(u,w)))).
P(i(u,i(i(u,v),v))).
P(i(u,i(i(i(v,v),i(u,w)),w))).
P(i(i(i(i(u,v),v),i(i(u,v),v)),i(u,i(i(u,v),v)))).
P(i(i(i(u,i(i(u,v),v)),w),w)).
P(i(i(i(u,u),i(i(v,i(i(i(w,w),i(v,x)),x)),y)),y)).
P(i(i(i(i(i(i(u,v),v),i(i(u,v),v)),i(u,i(i(u,v),v))),w),w)).
P(i(i(i(u,u),i(i(i(i(v,i(i(v,w),w)),x),x),y)),y)).
P(i(i(i(u,u),i(i(v,v),w)),w)).
P(i(u,u)). % I
P(i(i(u,i(v,w)),i(v,i(u,w)))). % C
P(i(u,i(i(i(v,v),i(i(i(i(w,i(i(w,x),x)),y),y),i(u,z))),z))).
P(i(i(u,v),i(i(v,w),i(u,w)))). % B'
P(i(i(u,i(v,w)),i(i(i(v,w),v),i(u,w)))).
P(i(i(i(i(u,v),i(w,v)),x),i(i(w,u),x))).
P(i(i(i(i(u,v),v),i(u,v)),i(u,v))).
P(i(i(u,i(u,v)),i(u,v))). % W
end_of_list.

list(usable).
-P(i(x,y)) | -P(x) | P(y).
-P(i(a,a)) | -P(i(i(a,i(b,c)),i(b,i(a,c)))) | -P(i(i(a,b),i(i(b,c),i(a,c)))) | -P(i(i(a,i(a,b)),i(a,b))) | $ANS(all).
end_of_list.

list(sos).
P(i(i(i(i(i(u,v),i(w,u)),i(i(u,v),i(w,v))),i(i(x,x),i(y,i(z,v6)))),i(i(v7,y),i(z,i(v7,v6))))).
end_of_list.

list(passive).
-P(i(a,a)) | $ANS(I).
-P(i(i(a,i(b,c)),i(b,i(a,c)))) | $ANS(C).
```

```
-P(i(i(a,b),i(i(b,c),i(a,c)))) | $ANS(B').
-P(i(i(a,i(a,b)),i(a,b))) | $ANS(W).
end_of_list.

list(demodulators).
% (i(i(x,x),y) = junk).
% (i(y,i(x,x)) = junk).
(i(x,junk) = junk).
(i(junk,x) = junk).
(P(junk) = $T).
end_of_list.
```

Veroff's hints strategy enables the researcher to guide the actions of an automated reasoning program. The formulas or equations, chosen by the user, that are placed in the hints list are intended to suggest that each, if deduced, is to be preferred for inference-rule initiation. The following two commands are usually included in my studies.

```
assign(bsub_hint_wt,1).
set(keep_hint_subsumers).
```

With these two commands, OTTER (or any program applying the hints strategy) will assign a weight (value) of 1 to any deduced conclusion that matches a hint, is subsumed by a hint, or subsumes a hint. Clauses assigned a weight of 1 are ordinarily given great preference for initiating the application of an inference rule in use.

Ancestor subsumption is an automated way to seek shorter proofs, although not guaranteed to do so. With this procedure, a second deduction of a retained clause is compared in the context of proof length. The strictly shorter derivation is that which is actively maintained.

The given input file found five proofs. The proofs of *I* and *B*' within the 20-step proof are each of length 7; the proof of *C* has length 9; the proof of *W* has length 19. The proof of the join of the four cited formulas has length 20. In other words, even with ancestor subsumption, no progress was made. I was quite surprised; yes, I was in fact disappointed.

So I decided to see what OTTER could find with a level-saturation approach. The second experiment used an input file quite like the preceding, but the following five lines appeared in it that were not in the file used for the just-described (first) experiment.

```
set(sos_queue).
assign(max_weight,40).
assign(change_limit_after,800).
assign(new_max_weight,16).
% assign(pick_given_ratio,1).
```

The presence of these five lines instructed the program to rely on level saturation (queue), to comment out the use of the ratio strategy (to be explained soon), to use the value of 40 for max_weight, and to change the value of max_weight to 16 after 800 clauses were chosen to initiate the application of an inference rule, in this case, condensed detachment. When the value *n* is assigned to pick_given_ratio, the program chooses for inference-rule application *n* clauses based on complexity, 1 based on first come first serve, *n*, 1, and the like. I thought that level saturation might prove effective in that the level of the proof of the join (found in the first experiment) was 12. I also thought that the program would not be able to examine all levels through level 12 without some lowering of max_weight and further lowering after, say, 800 clauses were used as given.

The second experiment yielded, in order, proofs of *I*, *I*, *I*, *B*', *C*, *W* and the join of the four targets. The respective lengths were 9, 8, 7, 7, 9, 13, and 22. The proof of the join, the last proof, has level 11. The job terminated because sos went empty, meaning no additional conclusions could be drawn within the given constraints.

Having obtained nothing that interested me (from the second experiment), I returned to the use of the given input file. In this third experiment, I used demodulation to block steps of the 20-step proof, one at a time, to see if the program could find a shorter proof. It worked. By blocking the fourth step, progress was made; then by also blocking the eighth step, more progress occurred. The following set of demodulators eventually was used.

```
list(demodulators).
(P(i(i(i(x,i(i(x,y),y)),z),z)) = junk).
(P(i(i(x,i(i(i(y,z),i(y,z)),i(i(u,u),i(y,z)))),i(y,i(x,z)))) = junk).
(i(x,junk) = junk).
(i(junk,x) = junk).
(P(junk) = $T).
end_of_list.
```

In order, proofs were completed of *B*', *C*, *I*, *W*, and the join. The respective lengths are 7, 10, 13, 13, and 17, the last proof (of the join) has level 12. Again you encounter that charming phenomenon of (in effect) trading shorter proofs of members of a conjunction for longer, resulting in a shorter proof of the conjunction (join). In particular, a 7-step proof of *I* is replaced with a 13-step proof, and a 9-step proof of *C* is replaced with a 10-step proof. You thus have yet more evidence that seeking shorter and ever shorter proofs is far from simple, and the most obvious approach often fails. Specifically, for example, when seeking a shorter proof of a conjunction, simply obtaining shorter and still shorter proofs of one of its members, although intuitively appealing, does in no way guarantee progress.

You might be curious about the differences in the proofs of the join, and, of course, you might wish to have in hand the 17-step proof (which I shall give very soon). The formulas (twenty of them) found in the proof of the join in the first experiment are, as expected, the twenty sent to me by Ulrich. Of the seventeen (deduced) formulas of the proof of the join found in the third experiment, the following four did not occur in the twenty.

```
P(i(i(i(i(x,y),y),z),i(x,z))).
P(i(i(i(x,x),i(i(y,z),i(u,y))),i(i(v,i(y,z)),i(u,i(v,z))))).
P(i(x,i(y,i(i(x,i(y,z)),z)))).
P(i(i(x,i(i(y,y),y)),i(x,y))).
```

A 17-Step Proof of the Join

```
----- Otter 3.3g-work, Jan 2005 -----
The process was started by wos on elephant.mcs.anl.gov,
Thu Oct 11 16:53:09 2007
The command was "otter". The process ID is 11438.

-----> EMPTY CLAUSE at   0.68 sec ----> 3657 [hyper,22,983,150,79,3563] $ANS(all).

Length of proof is 17. Level of proof is 12.

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

21 [] -P(i(x,y)) | -P(x) | P(y).
22 [] -P(i(a,a)) | -P(i(i(a,i(b,c)),i(b,i(a,c)))) | -P(i(i(a,b),i(i(b,c),i(a,c)))) | -P(i(i(a,i(a,b)),i(a,b))) | $ANS(all).
23 [] P(i(i(i(i(i(u,v),i(w,u)),i(i(u,v),i(w,v))),i(i(x,x),i(y,i(z,v6)))),i(i(v7,y),i(z,i(v7,v6))))).
33 [hyper,21,23,23] P(i(i(x,y),i(i(y,y),i(x,y)))).
35 [hyper,21,23,33] P(i(i(x,i(i(y,z),i(u,y))),i(i(y,z),i(x,i(u,z))))).
41 [hyper,21,23,35] P(i(i(x,i(i(y,y),i(z,y))),i(z,i(x,y)))).
47 [hyper,21,41,33] P(i(x,i(i(x,y),y))).
54 [hyper,21,33,47] P(i(i(i(i(x,y),y),i(i(x,y),y)),i(x,i(i(x,y),y)))).
```

68 [hyper,21,47,54] P(i(i(i(i(i(i(x,y),y),i(i(x,y),y)),i(x,i(i(x,y),y))),z),z)).
79 [hyper,21,23,68] P(i(i(x,y),i(i(y,z),i(x,z)))).
85 [hyper,21,79,79] P(i(i(i(i(x,y),i(z,y)),u),i(i(z,x),u))).
90 [hyper,21,79,47] P(i(i(i(i(x,y),y),z),i(x,z))).
131 [hyper,21,85,23] P(i(i(i(x,x),i(i(y,z),i(u,y))),i(i(v,i(y,z)),i(u,i(v,z))))).
150 [hyper,21,85,90] P(i(i(x,i(y,z)),i(y,i(x,z)))).
177 [hyper,21,90,150] P(i(x,i(y,i(i(x,i(y,z)),z)))).
395 [hyper,21,41,177] P(i(i(x,i(i(y,y),y)),i(x,y))).
983 [hyper,21,395,47] P(i(x,x)).
3331 [hyper,21,131,90] P(i(i(x,i(y,z)),i(i(i(y,z),y),i(x,z)))).
3443 [hyper,21,3331,47] P(i(i(i(i(x,y),y),i(x,y)),i(x,y))).
3563 [hyper,21,85,3443] P(i(i(x,i(x,y)),i(x,y))).

Before turning to the details concerning a proof of the Ulrich 35-symbol formula from (as hypotheses) *B*', *C*, *I*, and *W*, a few remarks are in order of a general nature. These observations may provide you with more insight into proof finding and proof shortening and, just perhaps, more about problem solving. I shall first focus on level saturation, an approach that, on the surface, might appear to be a panacea.

When relying on a level-saturation approach, the best approach is to allow the program to continue its search after having found a proof of a member, say *A*, of the conjunction under study. One important reason for doing so is that a longer proof of *A*, even if of higher level, may have more steps in common with the proofs of other members already proved. In other words, a second proof of *A*, though longer than a first proof, might share many or most of its steps with proofs already in hand of the other members of the conjunction in focus. Since, here, the object is to find a shorter proof of the conjunction, the length of the so-called new proof of the conjunction may be shorter than that in hand because of so many shared steps. Indeed, the length of the proof of a member is not the real concern, even if you can find a shorter proof of it.

This observation might suggest to you an approach for seeking shorter and still shorter proofs of, say, a conjunction. You could do the following, even write a program in this regard. First, you could have the program accrue a number, preferably more than one, of proofs of each member of the conjunction under study. Second, from each set of proofs, you or the program selects a proof. Third, you then compare the selected proofs pairwise or as an entire group with the objective of maximizing the number of shared proof steps. As may be overwhelmingly clear by now, the presence of some step in the shortest proof you can find for, say, *A* may interfere with finding a shorter proof of the conjunction. Indeed, that shortest proof may have many steps that are used nowhere else in the proof of the join, causing you or a program to return a so-called total proof of undesirable length.

Just in case you have not thought of it, a search for a proof, whether the goal is a conjunction or not, should continue after a first proof has been found if the goal is to find a shorter proof. The following example illustrates what can happen. A first proof can be found such that it consists of two chains of reasoning, each terminating with a level-4 conclusion. Each chain might be of length 4, deducing in order a conclusion of level 1, 2, 3, and 4. Such a proof has length 9. In some cases, if allowed to continue the search, a program (or a person) might find a proof that consists of (in order) a level-1 conclusion, a level-2, a level-3, a level-4, a level-5, and a level-6, the last being the desired result. This second proof has length 6.

Of course, I am not in any way implying that a level-saturation approach is of little or no use: quite the contrary. The difficulty with such a search, as I have observed repeatedly, is practicality: Even with constraints that appear quite restrictive, the size of the levels to be searched can grow wildly. Depending on the area in focus, an automated reasoning program can sometimes examine a decent number of levels thoroughly. In equivalential calculus, for example (as it is often studied to take advantage of what is called the 2-property), the levels do not grow as rapidly as with an area such as *RI*. The reason is that, under the stated conditions, all formulas in equivalential calculus have the property that each variable in a deduced conclusion occurs exactly twice. That property is clearly not present for *RI*. In one study of the Ulrich 35-symbol formula with the hypotheses consisting of *B*', *C*, *I*, and *W*, the size of level 0 (input) was 21 (clauses). On level 1 there were 7 additional clauses; on level 2 there were 76 additional clauses; on level 3 there were 1979 additional clauses; and on level 4 there were 152,495 additional clauses. The max_weight in this experiment was assigned the value 28; the goal was to deduce from the four given formulas the Ulrich 35-symbol formula. The experiment was terminated because of running out of memory, more than 800 megabytes. Of course, if the program were asked to search the space of deducible conclusions through even level 8, the task would have been out of reach.

As for the technique of seeing shorter proofs by using demodulation to block proof steps, obstacles are clearly encountered. For example, blocking the eighth step of some proof may produce no progress. But, after blocking the fourth, then blocking the eighth may produce progress; see the earlier discussion focusing on the fourth and eighth steps of a 20-step proof. Instead of blocking proof steps one at a time, one could have (with the use of a program written by McCune called otter-loopn, a bit different from his program otter-loop, which focuses on steps one at a time) blocked steps two at a time. If one had blocked two steps at a time in th example in focus, then the newer and shorter proof would have been found with the first attempt based on using demodulation to block proof steps. A quick review of the just-given observations when coupled with the following data strongly suggests that an algorithm for finding shorter proofs would be hard to find. In particular, in one experiment designed to find a proof of length strictly less than 17 (with the Ulrich 35-symbol formula as hypothesis), OTTER proved in order *I*, *B*', *C*, *W*, the join of the four, *W*, the join of the four, *I*, and the join the four. In order, the proof lengths were 8, 9, 7, 17, 19, 14, 20, 7, and 21. You see that, when OTTER found a second proof of *W* of length 14 to replace a 17-step proof, the proof of the resulting join went from 19 to 20. Then, when OTTER found a 7-step proof of *I* to replace an 8-step proof, the new proof of the join has length 21 rather than 20. Again, you see that a simple approach that focuses on finding shorter and still shorter proofs of members of a conjunction does not always lead to a shorter proof of the entire conjunction.

With the observations of a general nature complete for now, next in order is an account, including obstacles that were met and conquered, of how I attacked the problem of proving the Ulrich 35-symbol formula from the set of hypotheses consisting of *B*', *C*, *I*, and *W*.

3. A Potentially Formidable Task

The most obvious obstacle, and indeed a serious one, was the length or complexity of the target, the Ulrich 35-symbol formula. In the simplest terms, OTTER prefers (as researchers often do) short formulas or equations, ones with relatively few symbols in them. Now you might, having read various notebooks, naturally suggest the use of *resonators* (to be discussed almost immediately) or Veroff's hints. Specifically, you might consider including in an input file various formulas (in this case) that would quite likely appear in a proof of the Ulrich single axiom, and including symbol patterns whose shape would eventually direct the program toward the needed proof steps.

The objective of the *resonance strategy*, which relies on the use of *resonators*, is to enable the researcher to include (for OTTER, as weight templates) equations or formulas, called *resonators*, that will be used to direct a program's reasoning. A resonator is a symbol pattern (all of whose variables are considered indistinguishable) conjectured to merit special attention because it, and any matching expression, has particular appeal. To each resonator one assigns a value reflecting its relative (conjectured) significance: the smaller the value, the greater the significance. You could, for example, have in mind a set of steps that are promising in the context of finding a new proof or a first proof. In such an event, you might include each of those steps, as a resonator, to direct the program's reasoning accordingly.

To aid you in understanding what could be done, say, with hints, the following formula could be placed in the hints list with the notion that it, or those subsumed by it or subsuming it, could play a role in the sought-after proof.

P(i(i(i(i(i(i(x,y),y),i(i(x,y),y)),i(x,i(i(x,y),y))),z),z)).

Or, with some small assigned value, the formula could be placed (as a weight template) in the input file as a resonator. Further, perhaps some of its well-formed subexpressions could be included as weight templates with small values assigned to each. In that case, any deduced and retained conclusion would, if it contained a matching subexpression (where variables are treated as indistinguishable), have its weight adjusted based on the small value assigned to the subexpression. Summarizing, if good guesses or conjectures could be made about the nature of the sought-after proof, then corresponding advice (in the form of hints or resonators or weight templates for subexpressions) could be given (in the input file) to effectively direct the automated reasoning program toward the goal. In other words, you could (with the appropriate information) have the program treat complex formulas to be used as proof steps as being simple. Quite likely, one or more complex formulas, in addition to the Ulrich 35-symbol formula, would be needed in a proof that derives this fine single axiom.

From what I had in hand, I had no idea which hints or resonators or subexpressions were relevant to the desired proof. You might immediately wonder why this lack of knowledge did not prevent other studies from completing. Well, the best I can say is that the target was seldom if ever this complex (weight 35 not including the predicate symbol) or I had in hand a source of what appeared to be good hints or resonators. If you accept for the moment what I have just said, you might suggest a level-saturation approach. However, it would appear that max_weight must in that case be assigned the value 35. With such an assignment, not many levels could be traversed (as discussed earlier in this notebook). So, I asked myself, what could be done? And Ulrich provided a start, what he calls the building blocks of his 35-symbol formula, including the following.

```
i(i(i(u,v),i(w,u)),i(i(u,v),i(w,v)))
i(x,x)
```

Even before I learned of these two useful subexpressions, I conjectured that a good move would be to include a resonator, with a small assigned value, corresponding to the Ulrich axiom, the following.

```
weight(i(i(i(i(i(u,v),i(w,u)),i(i(u,v),i(w,v))),i(i(x,x),i(y,i(z,v6)))),i(i(v7,y),i(z,i(v7,v6)))),1).
% Ulrich 35-symbol formula
```

A glance at this single axiom shows that the preceding two building blocks are in fact well-formed subexpressions of it. And, because of factors I cannot identify, I conjectured that two additional subexpressions might be of use, the following.

```
i(i(x,x),i(i(y,i(z,v6)),i(i(v7,y),i(z,i(v7,v6)))))
i(i(y,i(z,v6)),i(i(v7,y),i(z,i(v7,v6))))
```

I thus had at least appealing to me, the following four weight templates to direct OTTER's search.

```
weight(i(i(i(i(i(u,v),i(w,u)),i(i(u,v),i(w,v))),i(i(x,x),i(y,i(z,v6)))),i(i(v7,y),i(z,i(v7,v6)))),1).
% Ulrich 35-symbol formula
weight(i(i(i(u,v),i(w,u)),i(i(u,v),i(w,v))),1).
weight(i(i(x,x),i(i(y,i(z,v6)),i(i(v7,y),i(z,i(v7,v6))))),1).
weight(i(i(y,i(z,v6)),i(i(v7,y),i(z,i(v7,v6)))),1).
```

I chose not to include the very simple building block in my first experiment. In addition, I did not know if any of the last three templates, viewed as theorems, was in fact a theorem. Even if none of the three turned out to be provable, each or all might aid the program by effectively directing its reasoning to key formulas to be used in the proof I was seeking. The input file I used in my first experiment in search of a deduction of the Ulrich 35-symbol formula (to prove it is a theorem of *RI*) from the now famous four-basis is the following.

An Input File for Seeking a Deduction of the Ulrich 35-Symbol Formula

```
set(hyper_res).
assign(max_mem,880000).
% assign(max_seconds,2).
% 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,1).
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).

weight_list(pick_and_purge).
weight(i(i(i(i(i(u,v),i(w,u)),i(i(u,v),i(w,v))),i(i(x,x),i(y,i(z,v6)))),i(i(v7,y),i(z,i(v7,v6)))),1).
% Ulrich 35-symbol formula
%  Following 3 were thought, at one time, to be subformulas of the Ulrich.
weight(i(i(i(u,v),i(w,u)),i(i(u,v),i(w,v))),1).
weight(i(i(x,x),i(i(y,i(z,v6)),i(i(v7,y),i(z,i(v7,v6))))),1).
weight(i(i(y,i(z,v6)),i(i(v7,y),i(z,i(v7,v6)))),1).
end_of_list.

list(usable).
-P(i(x,y)) | -P(x) | P(y).
end_of_list.

list(sos).
P(i(i(u,v),i(i(v,w),i(u,w)))). % B'
P(i(i(u,i(v,w)),i(v,i(u,w)))). % C
P(i(u,u)). % I
P(i(i(u,i(u,v)),i(u,v))). % W
end_of_list.

list(passive).
-P(i(i(i(i(i(a1,a2),i(a3,a1)),i(i(a1,a2),i(a3,a2))),i(i(a4,a4),i(a5,i(a6,a7)))),i(i(a8,a5),i(a6,i(a8,a7))))) |
  $ANS(U35).
%  Following 3 are negs of, what were thought to be, subformulas of the Ulrich.
-P(i(i(i(a1,b),i(a3,a1)),i(i(a1,b),i(a3,b)))) | $ANS(inter).
-P(i(i(a4,a4),i(i(a5,i(a6,b6)),i(i(b7,a5),i(a6,i(b7,b6)))))) | $ANS(inter).
-P(i(i(a5,i(a6,b6)),i(i(b7,a5),i(a6,i(b7,b6))))) | $ANS(inter).
-P(i(a,a)) | $ANS(I).
-P(i(i(a,i(b,c)),i(b,i(a,c)))) | $ANS(C).
-P(i(i(a,b),i(i(b,c),i(a,c)))) | $ANS(B').
-P(i(i(a,i(a,b)),i(a,b))) | $ANS(W).
end_of_list.


list(hints).
%  Following 20 derive B' C I and W from the new Ulrich 35-symbol single axiom for R.
P(i(i(u,v),i(i(v,v),i(u,v)))).
P(i(i(u,i(i(v,w),i(x,v))),i(i(v,w),i(u,i(x,w))))).
P(i(i(u,i(i(v,v),i(w,v))),i(w,i(u,v)))).
P(i(i(u,i(i(i(v,w),i(v,w)),i(i(x,x),i(v,w)))),i(v,i(u,w)))).
P(i(u,i(i(u,v),v))).
P(i(u,i(i(i(v,v),i(u,w)),w))).
P(i(i(i(i(u,v),v),i(i(u,v),v)),i(u,i(i(u,v),v)))).
```

```
P(i(i(i(u,i(i(u,v),v)),w),w)).
P(i(i(i(u,u),i(i(v,i(i(i(w,w),i(v,x)),x)),y)),y)).
P(i(i(i(i(i(i(u,v),v),i(i(u,v),v)),i(u,i(i(u,v),v))),w),w)).
P(i(i(i(u,u),i(i(i(i(v,i(i(v,w),w)),x),x),y)),y)).
P(i(i(i(u,u),i(i(v,v),w)),w)).
P(i(u,u)). % I
P(i(i(u,i(v,w)),i(v,i(u,w)))). % C
P(i(u,i(i(i(v,v),i(i(i(i(w,i(i(w,x),x)),y),y),i(u,z))),z))).
P(i(i(u,v),i(i(v,w),i(u,w)))). % b'
P(i(i(u,i(v,w)),i(i(i(v,w),v),i(u,w)))).
P(i(i(i(i(u,v),i(w,v)),x),i(i(w,u),x))).
P(i(i(i(i(u,v),v),i(u,v)),i(u,v))).
P(i(i(u,i(u,v)),i(u,v))). % W
end_of_list.
list(demodulators).
%  (P(i(i(i(x,i(i(x,y),y)),z),z)) = junk).
%  (P(i(i(x,i(i(i(y,z),i(y,z)),i(i(u,u),i(y,z))))),i(y,i(x,z)))) = junk).
%  (i(i(x,x),y) = junk).
%  (i(y,i(x,x)) = junk).
(i(x,junk) = junk).
(i(junk,x) = junk).
(P(junk) = $T).
```

With this input file, I sought of course to deduce the Ulrich formula. In that I strongly suspected that goal would be at least temporarily out of reach, I sought to find proofs of three so-called subformulas of the Ulrich formula, not knowing whether any or all are actually theorems of *RI*. As you see, I used a weight_list with items to direct OTTER's search. I placed in list(passive) the negations of the primary and secondary goals. I placed in list(sos) the famous four basis elements of *RI*, the key axiom system in hand (other than Ulrich's single axiom). I set ancestor subsumption, and I do not know why in that its presence slows the program substantially. With assign(max_proofs,-1), I encouraged the program to find as many proofs as various constraints allowed. Moreover, I included a hints list with the only formulas I thought might be of use in directing the reasoning, namely, proof steps that derive the 4-basis from the Ulrich single axiom.

In less than 2 CPU-seconds, OTTER proved the third so-called subformula (as a theorem) and then presented three proofs of the first of the three. The two proved formulas follow.

```
139 [hyper,32,43,48] P(i(i(x,i(y,z)),i(i(u,x),i(y,i(u,z))))).
5304 [hyper,32,269,104] P(i(i(i(x,y),i(z,x)),i(i(x,y),i(z,y)))).
```

The number of the clause as it is retained is in the first field of each of the two proved theorems. The proofs are of respective lengths 4, 8, 6, and 4. Perhaps the second of the three subformulas is itself not a theorem. Of course, as you might expect, the primary goal remained unproven. As you will see almost immediately (in Experiment 3), including ancestor subsumption was fortunate because it enabled the program to find three proofs of one of the secondary targets!

For the second experiment, I turned to level saturation by including the command set(sos_queue), of course commenting out pick_given_ratio. Very important, I included in list(sos) as (in effect) lemmas the two proved and just-cited formulas. For reasons I cannot recall, I continued to use ancestor subsumption and a max_weight of 72, and I did not reduce the max_weight at any point in the experiment. The following proof was found, of course a proof that depends on the two items proved in Experiment 1.

An Intermediate Proof of Value

```
----- Otter 3.3g-work, Jan 2005 -----
The process was started by wos on elephant.mcs.anl.gov,
```

Sun Oct 14 15:50:34 2007
The command was "otter". The process ID is 12786.

----> UNIT CONFLICT at 6018.18 sec ----> 307120 [binary,307119.1,21.1] $ANS(U35).

Length of proof is 7. Level of proof is 4.

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

21 [] -P(i(i(i(i(i(a1,a2),i(a3,a1)),i(i(a1,a2),i(a3,a2))),i(i(a4,a4),i(a5,i(a6,a7)))),i(i(a8,a5), i(a6,i(a8,a7))))) | $ANS(U35).
32 [] -P(i(x,y)) | -P(x) | P(y).
33 [] P(i(i(x,y),i(i(y,z),i(x,z)))).
35 [] P(i(i(x,i(y,z)),i(y,i(x,z)))).
37 [] P(i(x,x)).
41 [] P(i(i(x,i(y,z)),i(i(u,x),i(y,i(u,z))))).
43 [] P(i(i(i(x,y),i(z,x)),i(i(x,y),i(z,y)))).
46 [hyper,32,35,35] P(i(x,i(i(y,i(x,z)),i(y,z)))).
48 [hyper,32,35,33] P(i(i(x,y),i(i(z,x),i(z,y)))).
75 [hyper,32,46,43] P(i(i(x,i(i(i(y,z),i(u,y)),i(i(y,z),i(u,z))),v)),i(x,v))).
78 [hyper,32,46,37] P(i(i(x,i(i(y,y),z)),i(x,z))).
97 [hyper,32,48,41] P(i(i(x,i(y,i(z,u))),i(x,i(i(v,y),i(z,i(v,u)))))).
1055 [hyper,32,75,78] P(i(i(i(i(i(x,y),i(z,x)),i(i(x,y),i(z,y))),i(i(u,u),v)),v)).
307119 [hyper,32,97,1055] P(i(i(i(i(i(x,y),i(z,x)),i(i(x,y),i(z,y))),i(i(u,u),i(v,i(w,v6)))),i(i(v7,v), i(w,i(v7,v6))))).

As for the third experiment, the one I promised would show how fortune smiled when ancestor subsumption was included in the first experiment, I returned to the first experiment and its (given) input file. I immediately modified the input file by first taking all of the proof steps of the completed proofs in the first experiment (those of two of the three subformulas) and shorted them to remove duplicates. I obtained eleven formulas. Those, in addition to the seven formulas of the seven-step proof completed in the second experiment, I placed in the amended input file as hints. I continued, as in the two preceding experiments, to use the already-discussed weight templates. Those templates together with the eighteen hints would, I was almost certain, enable OTTER to return to me a deduction of the Ulrich single axiom, a proof based on the now familiar 4-basis. To remove any doubt, I did not include any lemmas, in particular, the two proved in the first experiment.

OTTER presented the following satisfying proof.

A Derivation of the Ulrich Single Axiom in RI

----- Otter 3.3g-work, Jan 2005 -----
The process was started by wos on elephant.mcs.anl.gov,
Sun Oct 14 18:18:43 2007
The command was "otter". The process ID is 15105.

----> UNIT CONFLICT at 0.25 sec ----> 1429 [binary,1428.1,39.1] $ANS(U35).

Length of proof is 14. Level of proof is 6.

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

39 [] -P(i(i(i(i(i(a1,a2),i(a3,a1)),i(i(a1,a2),i(a3,a2))),i(i(a4,a4),i(a5,i(a6,a7)))),i(i(a8,a5), i(a6,i(a8,a7))))) | $ANS(U35).

50 [] -P(i(x,y)) | -P(x) | P(y).
51 [] P(i(i(x,y),i(i(y,z),i(x,z)))).
53 [] P(i(i(x,i(y,z)),i(y,i(x,z)))).
55 [] P(i(x,x)).
57 [] P(i(i(x,i(x,y)),i(x,y))).
59 [hyper,50,51,51] P(i(i(i(i(x,y),i(z,y)),u),i(i(z,x),u))).
60 [hyper,50,53,53] P(i(x,i(i(y,i(x,z)),i(y,z)))).
61 [hyper,50,51,53] P(i(i(i(x,i(y,z)),u),i(i(y,i(x,z)),u))).
62 [hyper,50,53,51] P(i(i(x,y),i(i(z,x),i(z,y)))).
66 [hyper,50,59,59] P(i(i(x,i(y,z)),i(i(u,y),i(x,i(u,z))))).
77 [hyper,50,60,55] P(i(i(x,i(i(y,y),z)),i(x,z))).
96 [hyper,50,62,57] P(i(i(x,i(y,i(y,z))),i(x,i(y,z)))).
98 [hyper,50,62,51] P(i(i(x,i(y,z)),i(x,i(i(z,u),i(y,u))))).
113 [hyper,50,61,66] P(i(i(x,i(y,z)),i(i(u,x),i(y,i(u,z))))).
204 [hyper,50,96,98] P(i(i(i(x,y),i(z,x)),i(i(x,y),i(z,y)))).
336 [hyper,50,62,113] P(i(i(x,i(y,i(z,u))),i(x,i(i(v,y),i(z,i(v,u)))))).
517 [hyper,50,60,204] P(i(i(x,i(i(i(i(y,z),i(u,y)),i(i(y,z),i(u,z))),v)),i(x,v))).
1229 [hyper,50,517,77] P(i(i(i(i(i(x,y),i(z,x)),i(i(x,y),i(z,y))),i(i(u,u),v)),v)).
1428 [hyper,50,336,1229] P(i(i(i(i(i(x,y),i(z,x)),i(i(x,y),i(z,y))),i(i(u,u),i(v,i(w,v6)))),i(i(v7,v),
i(w,i(v7,v6))))).

I was indeed pleased, in two ways. Most important, I had in hand a proof of the Ulrich impressive 35-symbol single axiom. In fact, I now had the two proofs, each relying solely on condensed detachment, needed to show that the Ulrich 35-symbol formula is indeed a single axiom. Second, I had proceeded in a manner (to me) quite different from earlier studies. In particular, I had used a building block (supplied by Ulrich), and I had used so-called subformulas of his axiom. (I learned later that some of the three so-called subformulas are in fact not subformulas of the Ulrich axiom.) I, of course, was almost required to see whether I could improve upon the given 14-step proof.

As is so typical of my research, I immediately turned to demodulation, to block deduced steps of the 14-step proof one at a time. I relied upon (another of) McCune's programs otter-loop (not to be confused with otter-loopn) to automatically seek proofs, by blocking (in each of a series of runs) proof steps of a given proof one at a time if so ordered to do so, within whatever constraints were placed on it by the input file in use. Because the 14-step proof had been obtained so quickly, I placed a 2 CPU-second limit on each of the 14 runs, realizing that the last was useless in that no proof would be found without that step. After all, the last step is the Ulrich single axiom. So I simply relied on the input file that yielded the 14-step proof.

When the fifth step of the fourteen was blocked (by demodulating the corresponding formula to junk), OTTER found a 13-step level-6 proof, the following.

A 14-Step Proof Deducing the Ulrich Axiom

----- Otter 3.3g-work, Jan 2005 -----
The process was started by wos on lemma.mcs.anl.gov,
Sun Oct 14 19:49:03 2007
The command was "otter". The process ID is 24305.

----> UNIT CONFLICT at 0.58 sec ----> 3319 [binary,3318.1,44.1] $ANS(U35).

Length of proof is 13. Level of proof is 6.

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

39 [] -P(i(x,y)) | -P(x) | P(y).
40 [] P(i(i(u,v),i(i(v,w),i(u,w)))).
41 [] P(i(i(u,i(v,w)),i(v,i(u,w)))).
42 [] P(i(u,u)).
43 [] P(i(i(u,i(u,v)),i(u,v))).
44 [] -P(i(i(i(i(i(a1,a2),i(a3,a1)),i(i(a1,a2),i(a3,a2))),i(i(a4,a4),i(a5,i(a6,a7)))),i(i(a8,a5),
i(a6,i(a8,a7))))) | $ANS(U35).
57 [hyper,39,41,41] P(i(x,i(i(y,i(x,z)),i(y,z)))).
59 [hyper,39,41,40] P(i(i(x,y),i(i(z,x),i(z,y)))).
74 [hyper,39,57,42] P(i(i(x,i(i(y,y),z)),i(x,z))).
89 [hyper,39,40,59] P(i(i(i(i(x,y),i(x,z)),u),i(i(y,z),u))).
93 [hyper,39,59,43] P(i(i(x,i(y,i(y,z))),i(x,i(y,z)))).
94 [hyper,39,59,41] P(i(i(x,i(y,i(z,u))),i(x,i(z,i(y,u))))).
95 [hyper,39,59,40] P(i(i(x,i(y,z)),i(x,i(i(z,u),i(y,u))))).
182 [hyper,39,93,95] P(i(i(i(x,y),i(z,x)),i(i(x,y),i(z,y)))).
345 [hyper,39,57,182] P(i(i(x,i(i(i(i(y,z),i(u,y)),i(i(y,z),i(u,z))),v)),i(x,v))).
507 [hyper,39,345,74] P(i(i(i(i(i(x,y),i(z,x)),i(i(x,y),i(z,y))),i(i(u,u),v)),v)).
2619 [hyper,39,89,94] P(i(i(x,i(y,z)),i(i(u,x),i(y,i(u,z))))).
3046 [hyper,39,59,2619] P(i(i(x,i(y,i(z,u))),i(x,i(i(v,y),i(z,i(v,u)))))).
3318 [hyper,39,3046,507] P(i(i(i(i(i(x,y),i(z,x)),i(i(x,y),i(z,y))),i(i(u,u),i(v,i(w,v6)))),i(i(v7,v),
i(w,i(v7,v6))))).

Of the thirteen formulas in the just-cited proof, all but two are present (as formulas) in the earlier-cited 14-step proof. Also, two of the thirteen are not among the eighteen hints that were used to find this proof.

At this point, I wondered if I could find a proof of length strictly less than 13, a proof deriving the Ulrich single axiom. Because I now had a level-6 proof of the Ulrich single axiom, it seemed more than reasonable to see what would occur if I returned to the use of level saturation, which I did. I assigned max_weight the value 24 (to permit the program to examine a number of levels, perhaps even level 6), and I assigned max_distinct_var the value 8 because the Ulrich axiom relied on eight distinct variables. To avoid influencing OTTER too much, I included no hints but did still include the four weight templates present in these experiments. To give the program more latitude, I also commented out the single demodulator that had been used to block the use of a proof step of a 14-step proof. OTTER ran out of memory before a full examination of level 5; the last retained clause on level 4 was numbered 54231.

However, with the retention of clause (1018933) and more than 4740 CPU-seconds, the program found a 14-step level-5 proof of the Ulrich axiom. Of course, since the newer proof is of level 5 in contrast to the given level-6 proof, perhaps more gold could be mined. (By the way, this success showed that I could have found a proof deriving the Ulrich axiom from the 4-basis in one experiment, rather than the three discussed earlier. Although I did not think of it at the time, or not until writing this part of this notebook, this bit of evidence in part supports Overbeek's view that a good approach to proof seeking rests on assigning max_weight the value 1, 2, and the like; sometimes success will result sooner than expected.) The new 14-step proof relies on one formula not present in the earlier-cited 14-step proof. As an interesting aside, although some of the weight templates were mistakenly thought to be subformulas of the Ulrich axiom, their inclusion did indeed prove to be most useful. I—and now you—had in hand a harbinger of the future; indeed, subformulas can be put to good use, even if they are not taken from the target.

The obvious move was to return to the approach that had yielded the earlier-given 13-step proof, but with a few modifications. I assigned the max_weight the value 32 and the pick_given_ratio the value 1. These choices were intended to encourage OTTER to, just perhaps, insert new complex formulas. I retained the use of the demodulator whose presence had led to finding the cited 13-step proof from its predecessor of length 14. Most important, I included fourteen hints, those corresponding to the new 14-step proof, and no others. Success: McCune's fine program completed the following 13-step level-5 proof deriving the Ulrich single axiom.

A More Satisfying 13-Step Proof Deriving the Ulrich Axiom

----- Otter 3.3g-work, Jan 2005 -----
The process was started by wos on elephant.mcs.anl.gov,
Fri Oct 19 16:56:28 2007
The command was "otter". The process ID is 3026.

----> UNIT CONFLICT at 1.16 sec ----> 5028 [binary,5027.1,15.1] $ANS(U35).

Length of proof is 13. Level of proof is 5.

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

15 [] -P(i(i(i(i(i(a1,a2),i(a3,a1)),i(i(a1,a2),i(a3,a2))),i(i(a4,a4),i(a5,i(a6,a7)))),i(i(a8,a5),
i(a6,i(a8,a7))))) | $ANS(U35).
27 [] -P(i(x,y)) | -P(x) | P(y).
28 [] P(i(i(x,y),i(i(y,z),i(x,z)))).
30 [] P(i(i(x,i(y,z)),i(y,i(x,z)))).
32 [] P(i(x,x)).
34 [] P(i(i(x,i(x,y)),i(x,y))).
41 [hyper,27,30,30] P(i(x,i(i(y,i(x,z)),i(y,z)))).
44 [hyper,27,30,28] P(i(i(x,y),i(i(z,x),i(z,y)))).
50 [hyper,27,41,32] P(i(i(x,i(i(y,y),z)),i(x,z))).
83 [hyper,27,28,44] P(i(i(i(i(x,y),i(x,z)),u),i(i(y,z),u))).
88 [hyper,27,44,34] P(i(i(x,i(y,i(y,z))),i(x,i(y,z)))).
89 [hyper,27,44,30] P(i(i(x,i(y,i(z,u))),i(x,i(z,i(y,u))))).
90 [hyper,27,44,28] P(i(i(x,i(y,z)),i(x,i(i(z,u),i(y,u))))).
110 [hyper,27,30,50] P(i(x,i(i(x,i(i(y,y),z)),z))).
167 [hyper,27,88,90] P(i(i(i(x,y),i(z,x)),i(i(x,y),i(z,y)))).
254 [hyper,27,110,167] P(i(i(i(i(i(x,y),i(z,x)),i(i(x,y),i(z,y))),i(i(u,u),v)),v)).
4699 [hyper,27,83,89] P(i(i(x,i(y,z)),i(i(u,x),i(y,i(u,z))))).
4806 [hyper,27,44,4699] P(i(i(x,i(y,i(z,u))),i(x,i(i(v,y),i(z,i(v,u)))))).
5027 [hyper,27,4806,254] P(i(i(i(i(i(x,y),i(z,x)),i(i(x,y),i(z,y))),i(i(u,u),i(v,i(w,v6)))),i(i(v7,v),
i(w,i(v7,v6))))).

This new proof was the third proof of the Ulrich axiom in the experiment under discussion, the first two being of respective lengths 16 and 15. As noted, this new 13-step proof is indeed more satisfying in that its size (a concept introduced to me by Ulrich) is twelve less than the first 13-step proof. Size measures the total number of symbols (excluding commas and parentheses) in the set of deduced steps. In addition, but one difference exists between the two proofs in the context of the set of deduced steps—to me a surprising and piquant fact. In particular, of the following two formulas, the first is present in the new 13-step proof to replace the second from the first 13-step proof.

P(i(x,i(i(x,i(i(y,y),z)),z))).
P(i(i(x,i(i(i(i(y,z),i(u,y)),i(i(y,z),i(u,z))),v)),i(x,v))).

One set of experiments remained to conduct. These would focus on the given Rezus-style 93-symbol single axiom for *RI*. From a conversation with Ulrich, I learned that, most likely, no proof, based solely on condensed detachment, deriving the given Rezus-style axiom from the 4-basis in focus was available. How exciting to seek such a proof! The proposed expedition was especially intriguing in that OTTER (and I suspect most or all automated reasoning programs) has what amounts to a distaste for trying to complete a proof whose last step is a very, very complex item. As you will see in the next section, I already had the elements needed for a promising approach—although I was not aware of it at the time.

4. The Birth of a Strategy

New strategies sometimes are born because of the consideration or study of a single problem or theorem. In the case in focus, the theorem was a deduction, based solely on condensed detachment, from *B*', *C*, *I*, and *W* of the Ulrich new 35-symbol single axiom for *RI*. Ulrich knew his formula was a theorem of *RI* and had a condensed-detachment proof of that fact. However, my goal was to find such a proof with OTTER as the program and, of much importance, without knowledge gleaned from elsewhere. (Ulrich has his own program, which is interactive.) For example, I wished to avoid the use of resonators or hints. I knew the task might be difficult for OTTER because of its preference for smaller (less complex) formulas and equations. The program could easily become mired in formulas relying on fifteen or fewer symbols and never focus on expressions with, say, twenty or more symbols. So, what means could be devised to permit OTTER to cope well with complex formulas, as targets or as proof steps needed to reach such a target?

At this point, because of what you have seen, for example, in the input file for seeking a deduction of the Ulrich single axiom, you may have begun to conjecture as to the nature of the new strategy. The new strategy is called the *subformula strategy*. Its name signifies its nature, keying on subformulas of the target or theorem to be proved; for copious detail, see the notebook titled "The Subformula Strategy: Coping with Complex Expressions". With the Ulrich single axiom as the target, the following three expressions (each included in a weight template) were selected, each assigned a weight (value) of 1. (History requires I note that, although their inclusion proved valuable, only the first of the three corresponds to a subformula of the Ulrich single axiom; a later experiment focuses strictly on subformulas of the 35-symbol axiom and, therefore, corrects the given error.)

```
weight(i(i(i(u,v),i(w,u)),i(i(u,v),i(w,v))),1).
weight(i(i(x,x),i(i(y,i(z,v6)),i(i(v7,y),i(z,i(v7,v6))))),1).
weight(i(i(y,i(z,v6)),i(i(v7,y),i(z,i(v7,v6)))),1).
```

With the inclusion of these three templates, any deduced formula would, if it contained as a subexpression (not necessarily proper), a matching string with variables treated as indistinguishable, have its weight adjusted accordingly. For example, if the following formula was deduced, rather than assigning it a weight (priority or value) based on its symbol count, the formula would be assigned a weight of 2 because of matching the first of the three templates and because the predicate symbol is counted as having a value of 1.

```
P(i(i(i(u,v),i(w,u)),i(i(u,v),i(w,v)))).
```

For a second example, if a formula was deduced containing an expression, such as the following, that matched the first of the three cited templates, then that part of the deduced formula would be assigned a weight of 1.

```
i(i(i(x,y),i(x,x)),i(i(x,y),i(x,y)))
```

An inspection of the just-given subexpression, when compared with the first of the three cited weight templates, shows that it matches that first subexpression if you treat all variables as indistinguishable.

The following template was also present in the input file seeking a deduction of the Ulrich axiom, present to give great preference to the Ulrich axiom (if and when deduced) and to any deduced formula that matched it with all variables treated as indistinguishable.

```
weight(i(i(i(i(i(u,v),i(w,u)),i(i(u,v),i(w,v))),i(i(x,x),i(y,i(z,v6)))),i(i(v7,y),i(z,i(v7,v6)))),1).
% Ulrich 35-symbol formula
```

Yes, one other template could have been included that was not, a template that matches the Ulrich axiom but with a predicate symbol. It was not included simply because I did not think of the inclusion. I also included, perhaps rather hastily, twenty hints from a proof that deduces, from the Ulrich axiom, the 4-basis so prominent in this notebook. The rest of this story has been told earlier in this notebook. Indeed, in three experiments, the second and third building on its predecessor, the sought-after proof was in hand.

Immediately, you might raise two objections. First, you might justly say that all was not pure; after all, twenty hints were relied on from another related study. Those hints might have provided valuable, perhaps vital, direction of OTTER's reasoning. Second, you might wonder why I did not include additional subformulas. As to the second observation, I simply did not include such. I leave it to the curious to make the corresponding study. As for the first observation—a valid one indeed—later experimentation suggests that the twenty hints were relied on in a vital manner. With both objections clearly in view, the next test of the subformula strategy, which I will focus on shortly, will prove far more satisfying.

I pause here because it occurs to me that I can in fact perform one or more experiments that are pure, that rest solely for directing the reasoning on the subformula strategy, avoiding the use of hints and other weight templates. The following input file captures a successful experiment.

An Input File to Derive the Ulrich Single Axiom Relying Solely on the Subformula Strategy

```
set(hyper_res).
assign(max_mem,880000).
% assign(max_seconds,2).
% set(sos_queue).
assign(max_weight,8).
% assign(change_limit_after,1100).
% assign(new_max_weight,8).
assign(max_proofs,-1).
assign(pick_given_ratio,1).
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).

weight_list(pick_and_purge).
weight(i(i(i(i(i(u,v),i(w,u)),i(i(u,v),i(w,v))),i(i(x,x),i(y,i(z,v6)))),i(i(v7,y),i(z,i(v7,v6)))),1).
% Ulrich 35-symbol formula
% Following 8 from Beeson are subformulas of the Ulrich formula.
weight(i(v7,v6),1).
weight(i(i(v7,y),i(z,i(v7,v6))),1).
weight(i(y,i(z,v6)),1).
weight(i(i(x,x),i(y,i(z,v6))),1).
weight(i(i(u,v),i(w,v)),1).
weight(i(i(i(u,v),i(w,u)),i(i(u,v),i(w,v))),1).
weight(i(i(i(i(u,v),i(w,u)),i(i(u,v),i(w,v))),i(i(x,x),i(y,i(z,v6)))),1).
weight(i(i(i(i(i(u,v),i(w,u)),i(i(u,v),i(w,v))),i(i(x,x),i(y,i(z,v6)))),i(i(v7,y),i(z,i(v7,v6)))),1).
end_of_list.

list(usable).
-P(i(x,y)) | -P(x) | P(y).
end_of_list.

list(sos).
P(i(i(u,v),i(i(v,w),i(u,w)))). % B'
P(i(i(u,i(v,w)),i(v,i(u,w)))). % C
P(i(u,u)). % I
P(i(i(u,i(u,v)),i(u,v))). % W
end_of_list.
```

```
list(passive).
-P(i(i(i(i(i(a1,a2),i(a3,a1)),i(i(a1,a2),i(a3,a2))),i(i(a4,a4),i(a5,i(a6,a7)))),i(i(a8,a5),i(a6,i(a8,a7))))) |
  $ANS(U35).
% Following 3 are negs of, what were mistakenly treated as, subformulas of the Ulrich.
-P(i(i(i(a1,b),i(a3,a1)),i(i(a1,b),i(a3,b)))) | $ANS(inter).
-P(i(i(a4,a4),i(i(a5,i(a6,b6)),i(i(b7,a5),i(a6,i(b7,b6)))))) | $ANS(inter).
-P(i(i(a5,i(a6,b6)),i(i(b7,a5),i(a6,i(b7,b6))))) | $ANS(inter).
-P(i(a,a)) | $ANS(I).
-P(i(i(a,i(b,c)),i(b,i(a,c)))) | $ANS(C).
-P(i(i(a,b),i(i(b,c),i(a,c)))) | $ANS(B').
-P(i(i(a,i(a,b)),i(a,b))) | $ANS(W).
end_of_list.

list(demodulators).
% (P(i(i(i(x,i(i(x,y),y)),z),z)) = junk).
% (P(i(i(x,i(i(i(y,z),i(y,z)),i(i(u,u),i(y,z)))),i(y,i(x,z)))) = junk).
% (i(i(x,x),y) = junk).
% (i(y,i(x,x)) = junk).
(i(x,junk) = junk).
(i(junk,x) = junk).
(P(junk) = $T).
end_of_list.
```

With this input file, OTTER found in approximately 1652 CPU-seconds a 24-step level-8 proof of the Ulrich 35-symbol axiom, with retention of clause (104389).

As the just-described experiment was running, simultaneously I conducted another experiment but, rather than relying on complexity preference for choosing items to initiate applications of condensed detachment, relying on level saturation. The following two commands replaced their counterparts in the just-given input file.

```
set(sos_queue).
% assign(pick_given_ratio,1).
```

This second experiment also succeeded, returning a 17-step level-5 proof in approximately 43 CPU-seconds, with retention of clause (15434). The results of the two experiments give evidence for the power of the subformula strategy functioning unaided; indeed, no other guidance was provided. And now for a most pleasing study.

5. A Crucial Test

Enter Rezus—and, not counting a predicate symbol, his 93-symbol single axiom for *RI*, constructed for me by Ulrich.

```
% Following is a Rezus-style single axiom for RI.
P(i(i(i(x,i(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),w)),i(i(i(i(v6,v7),i(i(v7,v8),i(v6,v8))),
  i(i(i(i(v9,i(v9,v10)),i(v9,v10)),i(i(i(v11,v12),i(i(v12,v13),i(v11,v13))),v14)),v14),v15)),v15),
  (i(v16,i(i(i(v17,v17),i(i(v18,v18),i(i(v19,v19),i(i(v20,v20),i(v16,v21))))),v21)),v22))),v22)).
% Rezus-style
```

To be completely candid, I note that my goal was to rely solely on subformulas of the Rezus axiom and find some combinations of parameter values and set commands that would enable OTTER to complete a proof deriving the given Rezus-style formula from *B*', *C*, *I*, and *W*. Indeed, I intended to show that, other than obtaining by hand or by program subformulas of the given Rezus axiom and relying on some subset of them, no knowledge of any kind was required to derive the cited 93-symbol formula. If I succeeded, even if many experiments were required, then (at least from my viewpoint and that of Overbeek) impressive evidence would be in hand for the power of the subformula strategy. As for the objection that (in effect) the dice are being loaded in favor of this new strategy, I note that a program could be written to try numerous combinations.

One of the many experiments succeeded, and here is the input file that produced the desired proof.

An Input File for Deriving, from the 4-Basis, the Rezus Formula Solely with Subformulas

```
set(hyper_res).
assign(max_mem,880000).
% assign(max_seconds,3).
set(sos_queue).
set(input_sos_first).
assign(max_weight,12).
assign(change_limit_after,800).
assign(new_max_weight,8).
assign(max_proofs,-1).
assign(max_distinct_vars,25).
% assign(pick_given_ratio,2).
assign(bsub_hint_wt,1).
assign(report,5400).
set(keep_hint_subsumers).
% set(keep_hint_equivalents).
% set(ancestor_subsume).
set(back_sub).
set(order_history).
% set(process_input).
clear(print_kept).

weight_list(pick_and_purge).
% Following 20 are subformulas of the Rezus for RI, from Beeson.
weight(i(i(v6,v7),i(i(v7,v8),i(v6,v8))),1).
weight(i(i(z,z),i(i(u,u),i(i(v,v),i(x,w)))),1).
weight(i(i(v11,v12),i(i(v12,v13),i(v11,v13))),1).
weight(i(i(v19,v19),i(i(v21,v20),i(v16,v21))),0).
weight(i(i(i(v11,v12),i(i(v12,v13),i(v11,v13))),v14),1).
weight(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),1).
weight(i(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),w),1).
weight(i(i(v18,v18),i(i(v19,v19),i(i(v21,v20),i(v16,v21)))),0).
weight(i(x,i(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),w)),1).
weight(i(i(v17,v17),i(i(v18,v18),i(i(v19,v19),i(i(v21,v20),i(v16,v21))))),0).
weight(i(i(i(v17,v17),i(i(v18,v18),i(i(v19,v19),i(i(v21,v20),i(v16,v21))))),v21),0).
weight(i(i(i(v9,i(v9,v11)),i(v9,v10)),i(i(i(v11,v12),i(i(v12,v13),i(v11,v13))),v14)),0).
weight(i(v16,i(i(i(v17,v17),i(i(v18,v18),i(i(v19,v19),i(i(v21,v20),i(v16,v21))))),v21)),0).
weight(i(i(i(i(v9,i(v9,v11)),i(v9,v10)),i(i(i(v11,v12),i(i(v12,v13),i(v11,v13))),v14)),v14),0).
weight(i(i(v16,i(i(i(v17,v17),i(i(v18,v18),i(i(v19,v19),i(i(v21,v20),i(v16,v21))))),v21)),v22),0).
weight(i(i(i(i(i(v9,i(v9,v11)),i(v9,v10)),i(i(i(v11,v12),i(i(v12,v13),i(v11,v13))),v14)),v14),v15),0).
weight(i(i(i(v6,v7),i(i(v7,v8),i(v6,v8))),i(i(i(i(v9,i(v9,v11)),i(v9,v10)),i(i(i(v11,v12),
  i(i(v12,v13),i(v11,v13))),v14)),v14),v15)),0).
weight(i(i(i(i(v6,v7),i(i(v7,v8),i(v6,v8))),i(i(i(i(v9,i(v9,v11)),i(v9,v10)),i(i(i(v11,v12),i(i(v12,v13),
  i(v11,v13))),v14)),v14),v15)),v15),0).
weight(i(i(i(i(v6,v7),i(i(v7,v8),i(v6,v8))),i(i(i(i(v9,i(v9,v11)),i(v9,v10)),i(i(i(v11,v12),i(i(v12,v13),
  i(v11,v13))),v14)),v14),v15)),v15),i(i(v16,i(i(i(v17,v17),i(i(v18,v18),i(i(v19,v19),
  i(i(v20,v20),i(v16,v21))))),v21)),v22)),0).
```

```
weight(i(i(x,i(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),w)),i(i(i(i(v6,v7),i(i(v7,v8),i(v6,v8))),
  i(i(i(i(i(v9,i(v9,v11)),i(v9,v10)),i(i(i(v11,v12),i(i(v12,v13),i(v11,v13))),v14)),v14),v15)),v15),
  i(i(v16,i(i(i(v17,v17),i(i(v18,v18),i(i(v19,v19),i(i(v20,v20),i(v16,v21))))),v21)),v22))),0).
% % Following are building blocks for Rezus and Rezus itself.
% weight(i(i(x,y),i(i(y,z),i(x,z))),1).   % B'
% weight(i(x,i(i(i(y,y),i(x,z)),z)),1).   % Rezus's K1
% weight(i(x,i(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),w)),1). % Rezus's K4
weight(P(i(i(i(x,i(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),w)),i(i(i(i(v6,v7),i(i(v7,v8),i(v6,v8))),
  i(i(i(i(i(v9,i(v9,v10)),i(v9,v10)),i(i(i(v11,v12),i(i(v12,v13),i(v11,v13))),v14)),v14),v15)),v15),
  i(i(v16,i(i(i(v17,v17),i(i(v18,v18),i(i(v19,v19),i(i(v20,v20),i(v16,v21))))),v21)),v22))),v22)),1).
% A Rezus-style single axiom for RI
end_of_list.

list(usable).
-P(i(x,y)) | -P(x) | P(y).
% -P(i(a,a)) | -P(i(i(a,i(b,c)),i(b,i(a,c)))) | -P(i(i(a,b),i(i(b,c),i(a,c)))) | -P(i(i(a,i(a,b)),i(a,b))) | $ANS(all).
end_of_list.

list(sos).
% P(i(i(u,v),i(i(w,u),i(w,v)))). % B
P(i(i(u,v),i(i(v,w),i(u,w)))). % B'
P(i(i(u,i(v,w)),i(v,i(u,w)))). % C
P(i(u,u)). % I
P(i(i(u,i(u,v)),i(u,v))). % W
end_of_list.

list(passive).
% Following 20 are negs of subformulas of Rezus, from Beeson.
-P(i(i(b6,b7),i(i(b7,b8),i(b6,b8)))) | $ANS(inter4).
-P(i(i(a3,a3),i(i(a4,a4),i(i(b,b),i(a1,a6))))) | $ANS(inter4).
-P(i(i(b11,b12),i(i(b12,b13),i(b11,b13)))) | $ANS(inter4).
-P(i(i(b19,b19),i(i(b20,b20),i(b16,b21)))) | $ANS(inter4).
-P(i(i(i(b11,b12),i(i(b12,b13),i(b11,b13))),b14)) | $ANS(inter4).
-P(i(i(a2,a2),i(i(a3,a3),i(i(a4,a4),i(i(b,b),i(a1,a6)))))) | $ANS(inter4).
-P(i(i(i(a2,a2),i(i(a3,a3),i(i(a4,a4),i(i(b,b),i(a1,a6))))),a6)) | $ANS(inter4).
-P(i(i(b18,b18),i(i(b19,b19),i(i(b20,b20),i(b16,b21))))) | $ANS(inter4).
-P(i(a1,i(i(i(a2,a2),i(i(a3,a3),i(i(a4,a4),i(i(b,b),i(a1,a6))))),a6))) | $ANS(inter4).
-P(i(i(b17,b17),i(i(b18,b18),i(i(b19,b19),i(i(b20,b20),i(b16,b21)))))) | $ANS(inter4).
-P(i(i(i(b17,b17),i(i(b18,b18),i(i(b19,b19),i(i(b20,b20),i(b16,b21))))),b21)) | $ANS(inter4).
-P(i(i(i(b9,i(b9,b10)),i(b9,b10)),i(i(i(b11,b12),i(i(b12,b13),i(b11,b13))),b14))) | $ANS(inter4).
-P(i(b16,i(i(i(b17,b17),i(i(b18,b18),i(i(b19,b19),i(i(b20,b20),i(b16,b21))))),b21))) | $ANS(inter4).
-P(i(i(i(i(b9,i(b9,b10)),i(b9,b10)),i(i(i(b11,b12),i(i(b12,b13),i(b11,b13))),b14)),b14)) | $ANS(inter4).
-P(i(i(b16,i(i(i(b17,b17),i(i(b18,b18),i(i(b19,b19),i(i(b20,b20),i(b16,b21))))),b21)),b22)) | $ANS(inter4).
-P(i(i(i(i(i(b9,i(b9,b10)),i(b9,b10)),i(i(i(b11,b12),i(i(b12,b13),i(b11,b13))),b14)),b14),b15)) | $ANS(inter4).
-P(i(i(i(b6,b7),i(i(b7,b8),i(b6,b8))),i(i(i(i(i(b9,i(b9,b10)),i(b9,b10)),i(i(i(b11,b12),i(i(b12,b13),
  i(b11,b13))),b14)),b14),b15))) | $ANS(inter4).
-P(i(i(i(i(b6,b7),i(i(b7,b8),i(b6,b8))),i(i(i(i(i(b9,i(b9,b10)),i(b9,b10)),i(i(i(b11,b12),i(i(b12,b13),
  i(b11,b13))),b14)),b14),b15)),b15)) | $ANS(inter4).
-P(i(i(i(i(i(b6,b7),i(i(b7,b8),i(b6,b8))),i(i(i(i(i(b9,i(b9,b10)),i(b9,b10)),i(i(i(b11,b12),i(i(b12,b13),
  i(b11,b13))),b14)),b14),b15)),b15),i(i(b16,i(i(i(b17,b17),i(i(b18,b18),i(i(b19,b19),i(i(b20,b20),
  i(b16,b21))))),b21)),b22))) | $ANS(inter4).
-P(i(i(a1,i(i(i(a2,a2),i(i(a3,a3),i(i(a4,a4),i(i(b,b),i(a1,a6))))),a6)),i(i(i(i(b6,b7),i(i(b7,b8),i(b6,b8))),
  i(i(i(i(i(b9,i(b9,b10)),i(b9,b10)),i(i(i(b11,b12),i(i(b12,b13),i(b11,b13))),b14)),b14),b15)),b15),
```

```
  i(i(b16,i(i(i(b17,b17),i(i(b18,b18),i(i(b19,b19),i(i(b20,b20),i(b16,b21))))),b21)),b22)))) | $ANS(inter4).
% % Following are negs of B', K1, K4, and Rezus, the first 3 are building blocks for Rezus.
% -P(i(i(a1,a2),i(i(a2,a3),i(a1,a3)))) | $ANS(BP).
% -P(i(a1,i(i(i(a2,a2),i(a1,a3)),a3))) | $ANS(K1).
% -P(i(a1,i(i(i(a2,a2),i(i(a3,a3),i(i(a4,a4),i(i(a5,a5),i(a1,a6))))),a6))) | $ANS(K4).
-P(i(i(i(a1,i(i(i(a2,a2),i(i(a3,a3),i(i(a4,a4),i(i(b,b),i(a1,a6))))),a6)),i(i(i(i(b6,b7),i(i(b7,b8),i(b6,b8))),
  i(i(i(i(i(b9,i(b9,b10)),i(b9,b10)),i(i(i(b11,b12),i(i(b12,b13),i(b11,b13))),b14)),b14),b15)),b15),
  i(i(b16,i(i(i(b17,b17),i(i(b18,b18),i(i(b19,b19),i(i(b20,b20),i(b16,b21))))),b21)),b22))),b22)) |
  $ANS(REZ).
end_of_list.

list(demodulators).
% (i(i(x,x),y) = junk).
% (i(y,i(x,x)) = junk).
(i(x,junk) = junk).
(i(junk,x) = junk).
(P(junk) = $T).
end_of_list.
```

With the given input file, the following proof was completed.

Rezus Conquered by the Subformula Strategy, a Proof

```
----- Otter 3.3g-work, Jan 2005 -----
The process was started by wos on lemma.mcs.anl.gov,
Mon Apr 28 06:59:03 2008
The command was "otter". The process ID is 22582.

----> UNIT CONFLICT at 143645.03 sec ----> 671339 [binary,671338.1,66.1] $ANS(REZ).

Length of proof is 25. Level of proof is 9.

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

1 [] -P(i(x,y)) | -P(x) |P(y).
2 [] P(i(i(u,v),i(i(v,w),i(u,w)))).
3 [] P(i(i(u,i(v,w)),i(v,i(u,w)))).
4 [] P(i(u,u)).
5 [] P(i(i(u,i(u,v)),i(u,v))).
66 [] -P(i(i(i(a1,i(i(i(a2,a2),i(i(a3,a3),i(i(a4,a4),i(i(b,b),i(a1,a6))))),a6)),i(i(i(i(b6,b7),i(i(b7,b8),
  i(b6,b8))),i(i(i(i(i(b9,i(b9,b10)),i(b9,b10)),i(i(i(b11,b12),i(i(b12,b13),i(b11,b13))),b14)),b14),b15)),
  b15),i(i(b16,i(i(i(b17,b17),i(i(b18,b18),i(i(b19,b19),i(i(b20,b20),i(b16,b21))))),b21)),b22))),b22)) |
  $ANS(REZ).
70 [hyper,1,3,3] P(i(x,i(i(y,i(x,z)),i(y,z)))).
73 [hyper,1,3,2] P(i(i(x,y),i(i(z,x),i(z,y)))).
75 [hyper,1,3,4] P(i(x,i(i(x,y),y))).
78 [hyper,1,70,4] P(i(i(x,i(i(y,y),z)),i(x,z))).
81 [hyper,1,70,2] P(i(i(x,i(i(i(y,z),i(i(z,u),i(y,u))),v)),i(x,v))).
84 [hyper,1,70,73] P(i(i(x,i(i(i(y,z),i(i(u,y),i(u,z))),v)),i(x,v))).
88 [hyper,1,2,75] P(i(i(i(i(x,y),y),z),i(x,z))).
92 [hyper,1,75,2] P(i(i(i(i(x,y),i(i(y,z),i(x,z))),u),u)).
97 [hyper,1,78,78] P(i(i(i(x,x),i(i(y,y),z)),z)).
105 [hyper,1,3,81] P(i(x,i(i(x,i(i(i(y,z),i(i(z,u),i(y,u))),v)),v))).
```

110 [hyper,1,81,84] P(i(i(i(i(x,y),i(i(y,z),i(x,z))),i(i(i(u,v),i(i(w,u),i(w,v))),v6)),v6)).
131 [hyper,1,88,3] P(i(x,i(y,i(i(x,i(y,z)),z)))).
133 [hyper,1,88,2] P(i(x,i(i(y,z),i(i(x,y),z)))).
143 [hyper,1,3,92] P(i(x,i(i(i(i(y,z),i(i(z,u),i(y,u))),i(x,v)),v))).
148 [hyper,1,3,97] P(i(x,i(i(i(y,y),i(i(z,z),i(x,u))),u))).
192 [hyper,1,105,5] P(i(i(i(i(x,i(x,y)),i(x,y)),i(i(i(z,u),i(i(u,v),i(z,v))),w)),w)).
214 [hyper,1,105,110] P(i(i(i(i(i(i(x,y),i(i(y,z),i(x,z))),i(i(i(u,v),i(i(w,u),i(w,v))),v6)),v6),i(i(i(v7,v8), i(i(v8,v9),i(v7,v9))),v10)),v10)).
419 [hyper,1,143,148] P(i(i(i(i(x,y),i(i(y,z),i(x,z))),i(i(u,i(i(i(v,v),i(i(w,w),i(u,v6))),v6)),v7)),v7)).
637 [hyper,1,143,192] P(i(i(i(i(x,y),i(i(y,z),i(x,z))),i(i(i(i(i(u,i(u,v)),i(u,v)),i(i(i(w,v6),i(i(v6,v7), i(w,v7))),v8)),v8),v9)),v9)).
1545 [hyper,1,214,419] P(i(i(x,i(i(y,y),i(i(z,z),i(u,v)))),i(u,i(x,v)))).
3348 [hyper,1,131,637] P(i(x,i(i(i(i(i(i(y,z),i(i(z,u),i(y,u))),i(i(i(i(v,i(v,w)),i(v,w)),i(i(i(v6,v7), i(i(v7,v8),i(v6,v8))),v9)),v9),v10)),v10),i(x,v11)),v11))).
7257 [hyper,1,1545,1545] P(i(x,i(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),w))).
39452 [hyper,1,3348,7257] P(i(i(i(i(i(i(x,y),i(i(y,z),i(x,z))),i(i(i(i(u,i(u,v)),i(u,v)),i(i(i(w,v6), i(i(v6,v7),i(w,v7))),v8)),v8),v9)),v9),i(i(v10,i(i(i(v11,v11),i(i(v12,v12),i(i(v13,v13),i(i(v14,v14), i(v10,v15))))),v15)),v16)),v16)).
39641 [hyper,1,133,7257] P(i(i(x,y),i(i(i(z,i(i(i(u,u),i(i(v,v),i(i(w,w),i(i(v6,v6),i(z,v7))))),v7)),x),y))).
671338 [hyper,1,39641,39452] P(i(i(i(x,i(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),w)),i(i(i(i(i(v6,v7), i(i(v7,v8),i(v6,v8))),i(i(i(i(v9,i(v9,v10)),i(v9,v10)),i(i(i(v11,v12),i(i(v12,v13),i(v11,v13))),v14)), v14),v15)),v15),i(i(v16,i(i(i(v17,v17),i(i(v18,v18),i(i(v19,v19),i(i(v20,v20), i(v16,v21))))),v21)),v22))),v22)).

To prove that the Rezus-style formula is in fact a single axiom for *RI*, you must prove that it is a theorem of that area of logic, say, deducible from a known axiom system. You have in hand now an input file that accomplishes this task. Such a task ordinarily presents an obstacle most difficult to overcome. The difficulty rests with the fact that the Rezus formula is indeed complex, having weight 93 in symbol count if the predicate symbol is not counted. Then you must also prove that, from the Rezus formula, you can derive a known axiom system, for example, that consisting of *B*', *C*, *I*, and *W*.

This last assignment, on the surface, would appear to be straightforward and easy to complete. That was certainly my expectation. I thought I could simply rely on the twenty subformulas (given earlier in the input file for deriving Rezus), perhaps also rely on the four targets, and rather quickly obtain the desired proof that the Rezus formula implies a known axiom system.

How mistaken I was: failure after failure! I tried many experiments to complete the given assignment, some running out of memory, some running out of the set of support (meaning that nothing new could be deduced). Finally, I (in effect) cheated; indeed, I returned to earlier experiments some of which did yield a proof that the Rezus formula implies the 4-basis. Just for total clarity, none of those earlier experiments relied on the subformula strategy. The revisiting bore fruit, leading to the use of the following input file.

An Input File Relying on the Subformula Strategy to Deduce a 4-Basis from Rezus

```
set(hyper_res).
assign(max_mem,880000).
% assign(max_seconds,6).
set(sos_queue).
assign(max_weight,44).
assign(change_limit_after,100).
assign(new_max_weight,22).
assign(max_proofs,-1).
assign(max_distinct_vars,12).
```

```
% assign(pick_given_ratio,4).
assign(bsub_hint_wt,1).
set(keep_hint_subsumers).
% set(keep_hint_equivalents).
% set(ancestor_subsume).
set(back_sub).
set(order_history).
% set(process_input).
clear(print_kept).

weight_list(pick_and_purge).
% Following 20 are subformulas of the Rezus for RI, from Beeson.
weight(i(i(v6,v7),i(i(v7,v8),i(v6,v8))),1).
weight(i(i(z,z),i(i(u,u),i(i(v,v),i(x,w)))),1).
weight(i(i(v11,v12),i(i(v12,v13),i(v11,v13))),1).
weight(i(i(v19,v19),i(i(v21,v20),i(v16,v21))),1).
weight(i(i(i(v11,v12),i(i(v12,v13),i(v11,v13))),v14),1).
weight(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),1).
weight(i(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),w),1).
weight(i(i(v18,v18),i(i(v19,v19),i(i(v21,v20),i(v16,v21)))),1).
weight(i(x,i(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),w)),1).
weight(i(i(v17,v17),i(i(v18,v18),i(i(v19,v19),i(i(v21,v20),i(v16,v21))))),1).
weight(i(i(i(v17,v17),i(i(v18,v18),i(i(v19,v19),i(i(v21,v20),i(v16,v21))))),v21),1).
weight(i(i(i(v9,i(v9,v11)),i(v9,v10)),i(i(i(v11,v12),i(i(v12,v13),i(v11,v13))),v14)),1).
weight(i(v16,i(i(i(v17,v17),i(i(v18,v18),i(i(v19,v19),i(i(v21,v20),i(v16,v21))))),v21)),1).
weight(i(i(i(i(v9,i(v9,v11)),i(v9,v10)),i(i(i(v11,v12),i(i(v12,v13),i(v11,v13))),v14)),v14),1).
weight(i(i(v16,i(i(i(v17,v17),i(i(v18,v18),i(i(v19,v19),i(i(v21,v20),i(v16,v21))))),v21)),v22),1).
weight(i(i(i(i(i(v9,i(v9,v11)),i(v9,v10)),i(i(i(v11,v12),i(i(v12,v13),i(v11,v13))),v14)),v14),v15),1).
weight(i(i(i(v6,v7),i(i(v7,v8),i(v6,v8))),i(i(i(i(i(v9,i(v9,v11)),i(v9,v10)),i(i(i(v11,v12),i(i(v12,v13),
  i(v11,v13))),v14)),v14),v15)),1).
weight(i(i(i(i(v6,v7),i(i(v7,v8),i(v6,v8))),i(i(i(i(i(v9,i(v9,v11)),i(v9,v10)),i(i(i(v11,v12),
  i(i(v12,v13),i(v11,v13))),v14)),v14),v15)),v15),1).
weight(i(i(i(i(i(v6,v7),i(i(v7,v8),i(v6,v8))),i(i(i(i(i(v9,i(v9,v11)),i(v9,v10)),i(i(i(v11,v12),i(i(v12,v13),
  i(v11,v13))),v14)),v14),v15)),v15),i(i(v16,i(i(i(v17,v17),i(i(v18,v18),i(i(v19,v19),i(i(v20,v20),
  i(v16,v21))))),v21)),v22)),1).
weight(i(i(x,i(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),w)),i(i(i(i(i(v6,v7),i(i(v7,v8),i(v6,v8))),
  i(i(i(i(i(v9,i(v9,v11)),i(v9,v10)),i(i(i(v11,v12),i(i(v12,v13),i(v11,v13))),v14)),v14),v15)),v15),
  i(i(v16,i(i(i(v17,v17),i(i(v18,v18),i(i(v19,v19),i(i(v20,v20),i(v16,v21))))),v21)),v22))),1).
% Following are the targets, B' C I W.
weight(P(i(i(u,v),i(i(v,w),i(u,w)))),2). % B'
weight(P(i(i(u,i(v,w)),i(v,i(u,w)))),2). % C
weight(P(i(u,u)),2). % I
weight(P(i(i(u,i(u,v)),i(u,v))),2). % W
end_of_list.

list(usable).
-P(i(x,y)) | -P(x) | P(y).
-P(i(a,a)) | -P(i(i(a,b),i(i(b,c),i(a,c)))) | -P(i(i(a,i(b,c)),i(b,i(a,c)))) | -P(i(i(a,i(a,b)),i(a,b))) | $ANS(allBPCIW).
end_of_list.

list(sos).
% Following is the Rezus single axiom for RI; preceding is Ulrich's single.
P(i(i(i(x,i(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),w)),i(i(i(i(i(v6,v7),i(i(v7,v8),i(v6,v8))),
```

```
  i(i(i(i(i(v9,i(v9,v10)),i(v9,v10)),i(i(i(v11,v12),i(i(v12,v13),i(v11,v13))),v14)),v14),v15)),v15),
  i(i(v16,i(i(i(v17,v17),i(i(v18,v18),i(i(v19,v19),i(i(v20,v20),i(v16,v21))))),v21)),v22))),v22)).
% A Rezus-style single axiom for RI
end_of_list.

list(passive).
-P(i(a,a)) | $ANS(I).
-P(i(i(a,i(b,c)),i(b,i(a,c)))) | $ANS(C).
-P(i(i(a,b),i(i(b,c),i(a,c)))) | $ANS(BP).
-P(i(i(a,i(a,b)),i(a,b))) | $ANS(W).
end_of_list.

list(demodulators).
%  (i(i(x,x),y) = junk).
%  (i(y,i(x,x)) = junk).
(i(x,junk) = junk).
(i(junk,x) = junk).
(P(junk) = $T).
end_of_list.
```

To be precise, as a glance reveals, the subformula strategy was supplemented by the inclusion of weight templates (resonators, discussed earlier) for the four targets. (I have not conducted the appropriate experiment to see whether it was necessary to include the four templates corresponding to the four targets.)

With the given input file, OTTER succeeded—well, almost. Specifically, in order, OTTER proved *B*', *W*, *I*, and *C*. The proof of *C* was completed in approximately 128 CPU-seconds, with retention of clause (24975). A natural reaction amounts to asserting that the assignment was, after all, not s difficult. However, recall that a number of unsuccessful experiments were conducted before an appropriate set of parameter values and settings was found. As for the "almost", OTTER did not return a proof of the join of the four, even though it had proofs of the four individual members. The reason was that, in order to produce the desired proof (of the join), the last of the four (in this case) must be chosen to initiate an application of condensed detachment. In that level saturation was being used, many, many more formulas must be considered before *C* took center stage.

Because I preferred a proof of the join, two choices were available. First, I could take the proof steps of the four members of the join (found in the output of the level-saturation experiment) and add their correspondents as resonators. For that run—which now contained twenty-one new resonators, each assigned a value of 3—I would replace level saturation with a complexity-preference approach, assigning (as I did) the value 4 to the pick_given_ratio. The program is thus instructed to choose (for inference-rule initiation) 4 clauses by their weight, 1 by first come first serve, 4, 1, and the like. I chose 4 for the assigned value, rather than, say, 2, to encourage the use of conclusions that matched an included resonator.

The desired proof (of the join) was returned almost immediately, in less than 1 CPU-second with retention of clause (243). The second choice, which I also investigated, was to take the input file that I have given and comment out list(sos_queue) and comment in (by removing the per cent sign) the command assign(pick_given_ratio,4). In just over 4 CPU-seconds and with retention of clause (3218), OTTER returned a 19-step proof of level 10. I thus had an interesting contrast, in that the previously described experiment yielded a proof (of the join) of length 21 and level 8.

Yet one additional experiment merited conducting in the context of proving the 4-basis from Rezus. In that experiment, which relied on McCune's ratio strategy, I omitted the use of the resonators corresponding to the four targets to see what would occur. OTTER quickly proved each of the four members of the basis but did not return a proof of the join, even after a substantial amount of CPU time. The explanation rests with the fact that the program does not choose for inference-rule initiation the four members, one at a time, preventing it from completing a proof of the join. Therefore, when the goal is to prove the join of two or more elements, you are well advised to include weight templates (resonators) corresponding to the members to be proved. Further, to each, you had best assign a small value to encourage the program to choose, when and if proved, the various members of the join with the result of returning a proof of the join.

At this point, you might indeed wish to see appropriate proofs. Rather than giving those resulting from the just-described experiments, I shall instead give the shortest proofs known to me. I shall include brief comments about how these short proofs were obtained.

The shortest proof I have found to this date (May 4, 2008) for the theorem that asserts the deducibility of the Rezus formula from the 4-basis (*B*', *C*, *I*, and *W*) has length 15 and level 9, the following.

A 15-Step Proof for the Deducibility of Rezus from the 4-Basis

```
----- Otter 3.3g-work, Jan 2005 -----
The process was started by wos on lemma.mcs.anl.gov,
Sun May  4 16:51:56 2008
The command was "otter".  The process ID is 10502.

----> UNIT CONFLICT at   2.28 sec ----> 5229 [binary,5228.1,29.1] $ANS(REZ).

Length of proof is 15.  Level of proof is 9.

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

1 [] -P(i(x,y)) | -P(x) |P(y).
2 [] P(i(i(x,y),i(i(y,z),i(x,z)))).
3 [] P(i(i(u,i(v,w)),i(v,i(u,w)))).
4 [] P(i(u,u)).
5 [] P(i(i(u,i(u,v)),i(u,v))).
29 [] -P(i(i(i(a1,i(i(i(a2,a2),i(i(a3,a3),i(i(a4,a4),i(i(b,b),i(a1,a6))))),a6)),i(i(i(i(b6,b7),i(i(b7,b8),
  i(b6,b8))),i(i(i(i(b9,i(b9,b10)),i(b9,b10)),i(i(i(b11,b12),i(i(b12,b13),i(b11,b13))),b14)),b14),b15)),
  b15),i(i(b16,i(i(i(b17,b17),i(i(b18,b18),i(i(b19,b19),i(i(b20,b20),i(b16,b21))))),b21)),b22))),b22)) |
  $ANS(REZ).
34 [hyper,1,3,3] P(i(x,i(i(y,i(x,z)),i(y,z)))).
42 [hyper,1,34,5] P(i(i(x,i(i(i(y,i(y,z)),i(y,z)),u)),i(x,u))).
43 [hyper,1,34,4] P(i(i(x,i(i(y,y),z)),i(x,z))).
45 [hyper,1,34,2] P(i(i(x,i(i(i(y,z),i(i(z,u),i(y,u))),v)),i(x,v))).
76 [hyper,1,2,43] P(i(i(i(x,y),z),i(i(x,i(i(u,u),y)),z))).
94 [hyper,1,42,45] P(i(i(i(i(x,i(x,y)),i(x,y)),i(i(i(z,u),i(i(u,v),i(z,v))),w)),w)).
185 [hyper,1,34,94] P(i(i(x,i(i(i(i(y,i(y,z)),i(y,z)),i(i(i(u,v),i(i(v,w),i(u,w))),v6)),v6),v7)),i(x,v7))).
357 [hyper,1,45,185] P(i(i(i(i(x,y),i(i(y,z),i(x,z))),i(i(i(i(u,i(u,v)),i(u,v)),i(i(i(w,v6),i(i(v6,v7),
  i(w,v7))),v8)),v8),v9)),v9)).
581 [hyper,1,34,357] P(i(i(x,i(i(i(i(i(y,z),i(i(z,u),i(y,u))),i(i(i(i(v,i(v,w)),i(v,w)),i(i(i(v6,v7),
  i(i(v7,v8),i(v6,v8))),v9)),v9),v10)),v10),v11)),i(x,v11))).
4414 [hyper,1,76,3] P(i(i(x,i(i(y,y),i(z,u))),i(z,i(x,u)))).
4420 [hyper,1,76,4414] P(i(i(x,i(i(y,y),i(i(z,z),i(u,v)))),i(u,i(x,v)))).
4683 [hyper,1,4420,4420] P(i(x,i(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),w))).
4868 [hyper,1,34,4683] P(i(i(x,i(i(y,i(i(i(z,z),i(i(u,u),i(i(v,v),i(i(w,w),i(y,v6))))),v6)),v7)),i(x,v7))).
4948 [hyper,1,4868,581] P(i(i(i(x,i(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),w)),i(i(i(i(v6,v7),
  i(i(v7,v8),i(v6,v8))),i(i(i(i(v9,i(v9,v10)),i(v9,v10)),i(i(i(v11,v12),i(i(v12,v13),i(v11,v13))),
  v14)),v14),v15)),v15),v16)),v16)).
5228 [hyper,1,4868,4948] P(i(i(i(x,i(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),w)),i(i(i(i(v6,v7),
  i(i(v7,v8),i(v6,v8))),i(i(i(i(v9,i(v9,v10)),i(v9,v10)),i(i(i(v11,v12),i(i(v12,v13),i(v11,v13))),v14)),
  v14),v15)),v15),i(i(v16,i(i(i(v17,v17),i(i(v18,v18),i(i(v19,v19),i(i(v20,v20),i(v16,v21))))),v21)),
  v22))),v22)).
```

The approach I used (illustrated in an input file to follow) to discover the given 15-step proof was iterative, one run building on the results of an earlier run. Of course, various failures occurred along the way. When a shorter proof than that in hand was found, its proof steps as resonators were adjoined in a later attempt. the members of each new set of resonators were assigned a value smaller than the set that preceded it. I may have used demodulation blocking en route to the final result, blocking steps one at a time of the shortest proof in hand with the goal of finding a still shorter proof. I did use *cramming* on the path to finding the given 15-step proof, a strategy that I now touch on briefly.

The cramming strategy derives its name from its nature. In effect, the strategy focuses on a chosen subproof of the total proof in hand and then attempts to cram steps of the chosen subproof into new subproofs needed to complete a new and shorter total proof of the conjunction. The degree of success (measured in the decrease in total proof length) depends on how many deduced steps of the chosen subproof can be made to play double-duty, triple-duty, or more, in the sense that (for a given deduced step) it is used in two subproofs, three subproofs, or more. Alternatively, the cramming strategy derives its name from emphasizing the role of the subproof of one element (formula or equation) and attempting to find subproofs of the remaining elements such that their proof lengths can be (so-to-speak) crammed into a chosen length *j*. Typically, *j* is chosen to be less than the length of the shortest proof known for the theorem under study. In a typical case, the goal is to prove the conjunction of a set of formulas or equations. In that situation, you can choose the proof (subproof) of one of its members on which to cram. For the strategy, you take the steps of the chosen subproof and place them in (the initial) list(sos) and have the program attempt to force some or all of those steps into what is needed to complete the so-called total proof. Because of the basic mechanism that is employed by the strategy, level saturation, it seems almost certain that none of the masters of logic or mathematics applied such an approach, whereas cramming is indeed well suited to the actions of an automated reasoning program.

An Input File for Discovering a 15-Step Proof of Rezus from a 4-Basis

```
set(hyper_res).
assign(max_mem,880000).
% assign(max_seconds,2).
% set(sos_queue).
set(input_sos_first).
assign(max_weight,24).
% assign(change_limit_after,1500).
% assign(new_max_weight,32).
assign(max_proofs,-1).
assign(max_distinct_vars,25).
assign(pick_given_ratio,2).
assign(bsub_hint_wt,1).
set(keep_hint_subsumers).
% set(keep_hint_equivalents).
set(ancestor_subsume).
set(back_sub).
set(order_history).
% set(process_input).
clear(print_kept).

weight_list(pick_and_purge).
% Following 17/8, with 3 not in the 18/9, prove Rezus, temp.ulrich.singax.r.out4n19.
weight(P(i(x,i(i(y,i(x,z)),i(y,z)))),-7).
weight(P(i(i(x,i(y,i(z,u))),i(x,i(z,i(y,u))))),-7).
weight(P(i(i(x,y),i(i(y,z),i(x,z)))),-7).
weight(P(i(i(x,i(i(i(y,i(y,z)),i(y,z)),u)),i(x,u))),-7).
```

```
weight(P(i(i(x,i(i(y,y),z)),i(x,z))),-7).
weight(P(i(i(x,i(i(i(y,z),i(i(z,u),i(y,u))),v)),i(x,v))),-7).
weight(P(i(i(i(x,x),i(i(y,y),z)),z)),-7).
weight(P(i(i(x,i(i(y,y),i(z,u))),i(z,i(x,u)))),-7).
weight(P(i(i(i(x,x),i(i(y,y),i(i(z,z),u))),u)),-7).
weight(P(i(i(i(i(x,i(x,y)),i(x,y)),i(i(i(z,u),i(i(u,v),i(z,v))),w)),w)),-7).
weight(P(i(i(x,i(i(i(i(i(y,i(y,z)),i(y,z)),i(i(i(u,v),i(i(v,w),i(u,w))),v6)),v6),v7)),i(x,v7))),-7).
weight(P(i(i(i(i(x,y),i(i(y,z),i(x,z))),i(i(i(i(u,i(u,v)),i(u,v)),i(i(i(w,v6),i(i(v6,v7),
  i(w,v7))),v8)),v8),v9)),v9)),-7).
weight(P(i(i(x,i(i(i(i(i(y,z),i(i(z,u),i(y,u))),i(i(i(i(v,i(v,w)),i(v,w)),i(i(i(v6,v7),i(i(v7,v8),
  i(v6,v8))),v9)),v9),v10)),v10),v11)),i(x,v11))),-7).
weight(P(i(x,i(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),w))),-7).
weight(P(i(i(x,i(i(y,i(i(i(z,z),i(i(u,u),i(i(v,v),i(i(w,w),i(y,v6))))),v6)),v7)),i(x,v7))),-7).
weight(P(i(i(i(x,i(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),w)),i(i(i(i(i(v6,v7),
  i(i(v7,v8),i(v6,v8))),i(i(i(i(v9,i(v9,v10)),i(v9,v10)),i(i(i(v11,v12),i(i(v12,v13),
  i(v11,v13))),v14)),v14),v15)),v15),v16)),v16)),-7).
weight(P(i(i(i(x,i(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),w)),i(i(i(i(i(v6,v7),i(i(v7,v8),i(v6,v8))),
  i(i(i(i(v9,i(v9,v10)),i(v9,v10)),i(i(i(v11,v12),i(i(v12,v13),i(v11,v13))),v14)),v14),v15)),v15),
  i(i(v16,i(i(i(v17,v17),i(i(v18,v18),i(i(v19,v19),i(i(v20,v20),i(v16,v21))))),v21)),v22))),v22)),-7).
% Following 18/9 prove Rezus, temp.ulrich.singax.r.out4n10.
weight(P(i(x,i(i(y,i(x,z)),i(y,z)))),-6).
weight(P(i(i(x,i(y,i(z,u))),i(x,i(z,i(y,u))))),-6).
weight(P(i(i(x,y),i(i(y,z),i(x,z)))),-6).
weight(P(i(i(x,i(i(y,y),z)),i(x,z))),-6).
weight(P(i(i(x,i(i(i(y,z),i(i(z,u),i(y,u))),v)),i(x,v))),-6).
weight(P(i(i(i(i(x,y),i(x,z)),u),i(i(y,z),u))),-6).
weight(P(i(i(i(x,x),i(i(y,y),z)),z)),-6).
weight(P(i(x,i(i(x,i(i(i(y,z),i(i(z,u),i(y,u))),v)),v))),-6).
weight(P(i(i(x,i(y,z)),i(i(u,x),i(y,i(u,z))))),-6).
weight(P(i(i(i(i(x,i(x,y)),i(x,y)),i(i(i(z,u),i(i(u,v),i(z,v))),w)),w)),-6).
weight(P(i(i(x,i(i(y,y),i(i(z,z),i(u,v)))),i(u,i(x,v)))),-6).
weight(P(i(i(x,i(i(i(i(i(y,i(y,z)),i(y,z)),i(i(i(u,v),i(i(v,w),i(u,w))),v6)),v6),v7)),i(x,v7))),-6).
weight(P(i(x,i(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),w))),-6).
weight(P(i(i(i(i(x,y),i(i(y,z),i(x,z))),i(i(i(i(u,i(u,v)),i(u,v)),i(i(i(w,v6),i(i(v6,v7),
  i(w,v7))),v8)),v8),v9)),v9)),-6).
weight(P(i(i(x,i(i(y,i(i(i(z,z),i(i(u,u),i(i(v,v),i(i(w,w),i(y,v6))))),v6)),v7)),i(x,v7))),-6).
weight(P(i(i(x,i(i(i(i(i(y,z),i(i(z,u),i(y,u))),i(i(i(i(v,i(v,w)),i(v,w)),i(i(i(v6,v7),i(i(v7,v8),
  i(v6,v8))),v9)),v9),v10)),v10),v11)),i(x,v11))),-6).
weight(P(i(i(i(x,i(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),w)),i(i(i(i(i(v6,v7),i(i(v7,v8),
 i(v6,v8))),i(i(i(i(v9,i(v9,v10)),i(v9,v10)),i(i(i(v11,v12),i(i(v12,v13),i(v11,v13))),v14)),v14),v15))
  ,v15),v16)),v16)),-6).
weight(P(i(i(i(x,i(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),w)),i(i(i(i(i(v6,v7),i(i(v7,v8),
  i(v6,v8))),i(i(i(i(v9,i(v9,v10)),i(v9,v10)),i(i(i(v11,v12),i(i(v12,v13),i(v11,v13))),v14)),v14),v15)),v15),
  (i(v16,i(i(i(v17,v17),i(i(v18,v18),i(i(v19,v19),i(i(v20,v20),i(v16,v21))))),v21)),v22))),v22)),-6).
% Following 19/9 prove Rezus, 5 not in earlier 19, temp.ulrich.singax.r.out4n8.
weight(P(i(x,i(i(y,i(x,z)),i(y,z)))),-5).
weight(P(i(i(x,i(y,i(z,u))),i(x,i(z,i(y,u))))),-5).
weight(P(i(i(x,y),i(i(y,z),i(x,z)))),-5).
weight(P(i(i(x,i(i(y,y),z)),i(x,z))),-5).
weight(P(i(i(x,i(i(i(y,z),i(i(z,u),i(y,u))),v)),i(x,v))),-5).
weight(P(i(i(i(i(x,y),i(x,z)),u),i(i(y,z),u))),-5).
weight(P(i(i(i(x,x),i(i(y,y),z)),z)),-5).
weight(P(i(x,i(i(x,i(i(i(y,z),i(i(z,u),i(y,u))),v)),v))),-5).
```

weight(P(i(i(x,i(y,z)),i(i(u,x),i(y,i(u,z))))),-5).
weight(P(i(i(i(i(x,i(x,y)),i(x,y)),i(i(i(z,u),i(i(u,v),i(z,v))),w)),w)),-5).
weight(P(i(i(x,i(i(y,y),i(i(z,z),i(u,v)))),i(u,i(x,v)))),-5).
weight(P(i(i(x,i(i(i(i(i(y,i(y,z)),i(y,z)),i(i(i(u,v),i(i(v,w),i(u,w))),v6)),v6),v7)),i(x,v7))),-5).
weight(P(i(x,i(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),w))),-5).
weight(P(i(i(i(i(i(x,y),i(i(y,z),i(x,z))),i(i(i(i(u,i(u,v)),i(u,v)),i(i(i(w,v6),i(i(v6,v7),i(w,v7))),v8)),v8),v9)),v9)),-5).
weight(P(i(i(x,i(i(y,i(i(i(z,z),i(i(u,u),i(i(v,v),i(i(w,w),i(y,v6))))),v6)),v7)),i(x,v7))),-5).
weight(P(i(i(x,i(i(i(i(i(y,z),i(i(z,u),i(y,u))),i(i(i(i(v,i(v,w)),i(v,w)),i(i(i(v6,v7),i(i(v7,v8),
i(v6,v8))),v9)),v9),v10)),v10),v11)),i(x,v11))),-5).
weight(P(i(i(x,y),i(i(i(z,i(i(i(u,u),i(i(v,v),i(i(w,w),i(i(v6,v6),i(z,v7))))),v7)),x),y))),-5).
weight(P(i(i(i(x,i(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),w)),i(v6,i(i(v7,i(i(i(v8,v8),i(i(v9,v9),
i(i(v10,v10),i(i(v11,v11),i(v7,v12))))),v12)),v13))),i(v6,v13))),-5).
weight(P(i(i(i(x,i(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),w)),i(i(i(i(i(v6,v7),i(i(v7,v8),
i(v6,v8))),i(i(i(i(v9,i(v9,v10)),i(v9,v10)),i(i(i(v11,v12),i(i(v12,v13),i(v11,v13))),v14)),v14),v15)),v15),
i(i(v16,i(i(i(v17,v17),i(i(v18,v18),i(i(v19,v19),i(i(v20,v20),i(v16,v21))))),v21)),v22))),v22)),-5).
% Following 20/9 prove Rezus, temp.ulrich.singax.r.out4n6.
weight(P(i(x,i(i(y,i(x,z)),i(y,z)))),-4).
weight(P(i(i(x,i(y,i(z,u))),i(x,i(z,i(y,u))))),-4).
weight(P(i(i(x,y),i(i(y,z),i(x,z)))),-4).
weight(P(i(x,i(i(x,y),y))),-4).
weight(P(i(i(x,i(i(y,y),z)),i(x,z))),-4).
weight(P(i(i(x,i(i(i(y,z),i(i(z,u),i(y,u))),v)),i(x,v))),-4).
weight(P(i(i(i(i(x,y),i(x,z)),u),i(i(y,z),u))),-4).
weight(P(i(i(i(x,x),i(i(y,y),z)),z)),-4).
weight(P(i(x,i(i(x,i(i(i(y,z),i(i(z,u),i(y,u))),v)),v))),-4).
weight(P(i(i(x,i(y,z)),i(i(u,x),i(y,i(u,z))))),-4).
weight(P(i(i(i(i(x,i(x,y)),i(x,y)),i(i(i(z,u),i(i(u,v),i(z,v))),w)),w)),-4).
weight(P(i(i(x,i(i(y,y),i(i(z,z),i(u,v)))),i(u,i(x,v)))),-4).
weight(P(i(i(x,i(i(i(i(i(y,i(y,z)),i(y,z)),i(i(i(u,v),i(i(v,w),i(u,w))),v6)),v6),v7)),i(x,v7))),-4).
weight(P(i(x,i(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),w))),-4).
weight(P(i(i(x,i(i(y,i(i(i(z,z),i(i(u,u),i(i(v,v),i(i(w,w),i(y,v6))))),v6)),v7)),i(x,v7))),-4).
weight(P(i(i(x,y),i(i(i(z,i(i(i(u,u),i(i(v,v),i(i(w,w),i(i(v6,v6),i(z,v7))))),v7)),x),y))),-4).
weight(P(i(i(i(i(i(x,y),i(i(y,z),i(x,z))),i(i(i(i(u,i(u,v)),i(u,v)),i(i(i(w,v6),i(i(v6,v7),i(w,v7))),v8)),v8),v9)),v9)),-4).
weight(P(i(i(i(i(i(i(i(x,y),i(i(y,z),i(x,z))),i(i(i(i(u,i(u,v)),i(u,v)),i(i(i(w,v6),i(i(v6,v7),
i(w,v7))),v8)),v8),v9)),v9),v10),v10)),-4).
weight(P(i(i(i(i(i(i(i(x,y),i(i(y,z),i(x,z))),i(i(i(i(u,i(u,v)),i(u,v)),i(i(i(w,v6),i(i(v6,v7),i(w,v7))),v8)),v8),v9)),v9),
i(i(v10,i(i(i(v11,v11),i(i(v12,v12),i(i(v13,v13),i(i(v14,v14),i(v10,v15))))),v15)),v16)),v16)),-4).
weight(P(i(i(i(x,i(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),w)),i(i(i(i(i(v6,v7),i(i(v7,v8),i(v6,v8))),
i(i(i(i(v9,i(v9,v10)),i(v9,v10)),i(i(i(v11,v12),i(i(v12,v13),i(v11,v13))),v14)),v14),v15)),v15),
i(i(v16,i(i(i(v17,v17),i(i(v18,v18),i(i(v19,v19),i(i(v20,v20),i(v16,v21))))),v21)),v22))),v22)),-4).
% Following 21/8 prove Rezus, temp.ulrich.singax.r.out4n5.
weight(P(i(x,i(i(y,i(x,z)),i(y,z)))),-3).
weight(P(i(i(x,i(y,i(z,u))),i(x,i(z,i(y,u))))),-3).
weight(P(i(i(x,y),i(i(y,z),i(x,z)))),-3).
weight(P(i(x,i(i(x,y),y))),-3).
weight(P(i(i(x,i(i(y,y),z)),i(x,z))),-3).
weight(P(i(i(x,i(i(i(y,z),i(i(z,u),i(y,u))),v)),i(x,v))),-3).
weight(P(i(i(i(i(x,y),i(x,z)),u),i(i(y,z),u))),-3).
weight(P(i(i(i(i(x,y),i(i(y,z),i(x,z))),u),u)),-3).
weight(P(i(x,i(i(x,i(i(i(y,z),i(i(z,u),i(y,u))),v)),v))),-3).
weight(P(i(i(x,i(y,z)),i(i(u,x),i(y,i(u,z))))),-3).
weight(P(i(x,i(i(i(i(y,z),i(i(z,u),i(y,u))),i(x,v)),v))),-3).
weight(P(i(i(i(x,x),i(i(y,y),z)),z)),-3).

weight(P(i(i(x,i(i(y,y),i(i(z,z),i(u,v)))),i(u,i(x,v)))),-3).
weight(P(i(x,i(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),w))),-3).
weight(P(i(i(x,i(i(y,i(i(i(z,z),i(i(u,u),i(i(v,v),i(i(w,w),i(y,v6))))),v6)),v7)),i(x,v7))),-3).
weight(P(i(i(x,y),i(i(i(z,i(i(i(u,u),i(i(v,v),i(i(w,w),i(i(v6,v6),i(z,v7))))),v7)),x),y))),-3).
weight(P(i(i(i(i(x,i(x,y)),i(x,y)),i(i(i(z,u),i(i(u,v),i(z,v))),w)),w)),-3).
weight(P(i(i(i(i(x,y),i(i(y,z),i(x,z))),i(i(i(i(i(u,i(u,v)),i(u,v)),i(i(i(w,v6),i(i(v6,v7),i(w,v7))),v8)),v8),v9)),v9)),-3).
weight(P(i(i(i(i(i(i(x,y),i(i(y,z),i(x,z))),i(i(i(i(i(u,i(u,v)),i(u,v)),i(i(i(w,v6),i(i(v6,v7),
i(w,v7))),v8)),v8),v9)),v9),v10),v10)),-3).
weight(P(i(i(i(i(i(i(x,y),i(i(y,z),i(x,z))),i(i(i(i(i(u,i(u,v)),i(u,v)),i(i(i(w,v6),i(i(v6,v7),
i(w,v7))),v8)),v8),v9)),v9),i(i(v10,i(i(i(v11,v11),i(i(v12,v12),i(i(v13,v13),i(i(v14,v14),
i(v10,v15))))),v15)),v16)),v16)),-3).
weight(P(i(i(i(x,i(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),w)),i(i(i(i(v6,v7),i(i(v7,v8),i(v6,v8))),
i(i(i(i(v9,i(v9,v10)),i(v9,v10)),i(i(i(v11,v12),i(i(v12,v13),i(v11,v13))),v14)),v14),v15)),v15),
i(i(v16,i(i(i(v17,v17),i(i(v18,v18),i(i(v19,v19),i(i(v20,v20),i(v16,v21))))),v21)),v22))),v22)),-3).
% Following 6, by cramming on the later 24, complete a proof of Rezus, temp.ulrich.singax.r.out4n4.
weight(P(i(x,i(i(y,z),i(i(x,y),z)))),-2).
weight(P(i(i(i(i(i(i(x,y),i(i(y,z),i(x,z))),i(i(i(i(i(u,i(u,v)),i(u,v)),i(i(i(w,v6),i(i(v6,v7),
i(w,v7))),v8)),v8),v9)),v9),v10),v10)),-2).
weight(P(i(i(x,i(i(y,i(i(i(z,z),i(i(u,u),i(i(v,v),i(i(w,w),i(y,v6))))),v6)),v7)),i(x,v7))),-2).
weight(P(i(i(x,y),i(i(i(z,i(i(i(u,u),i(i(v,v),i(i(w,w),i(i(v6,v6),i(z,v7))))),v7)),x),y))),-2).
weight(P(i(i(i(i(i(i(x,y),i(i(y,z),i(x,z))),i(i(i(i(i(u,i(u,v)),i(u,v)),i(i(i(w,v6),i(i(v6,v7),
i(w,v7))),v8)),v8),v9)),v9),i(i(v10,i(i(i(v11,v11),i(i(v12,v12),i(i(v13,v13),i(i(v14,v14),
i(v10,v15))))),v15)),v16)),v16)),-2).
weight(P(i(i(i(x,i(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),w)),i(i(i(i(v6,v7),i(i(v7,v8),i(v6,v8))),
i(i(i(i(v9,i(v9,v10)),i(v9,v10)),i(i(i(v11,v12),i(i(v12,v13),i(v11,v13))),v14)),v14),v15)),v15),
i(i(v16,i(i(i(v17,v17),i(i(v18,v18),i(i(v19,v19),i(i(v20,v20),i(v16,v21))))),v21)),v22))),v22)),-2).
% Following 24 shorted proof steps of subformulas and building blocks, temp.ulrich.r.out4n1.
weight(P(i(i(i(i(i(i(x,i(x,y)),i(x,y)),i(i(i(z,u),i(i(u,v),i(z,v))),w)),w),v6),v6)),-1).
weight(P(i(i(i(i(x,i(x,y)),i(x,y)),i(i(i(z,u),i(i(u,v),i(z,v))),w)),w)),-1).
weight(P(i(i(i(i(x,i(x,y)),i(x,y)),z),z)),-1).
weight(P(i(i(i(i(x,y),i(i(y,z),i(x,z))),i(i(i(i(i(u,i(u,v)),i(u,v)),i(i(i(w,v6),i(i(v6,v7),
i(w,v7))),v8)),v8),v9)),v9)),-1).
weight(P(i(i(i(i(x,y),i(i(y,z),i(x,z))),u),u)),-1).
weight(P(i(i(i(i(x,y),i(x,z)),u),i(i(y,z),u))),-1).
weight(P(i(i(i(i(x,y),i(z,y)),u),i(i(z,x),u))),-1).
weight(P(i(i(i(x,x),i(i(y,y),z)),z)),-1).
weight(P(i(i(i(x,x),y),y)),-1).
weight(P(i(i(i(x,y),i(z,x)),i(i(x,y),i(z,y)))),-1).
weight(P(i(i(x,i(i(i(y,z),i(i(z,u),i(y,u))),v)),i(x,v))),-1).
weight(P(i(i(x,i(i(y,y),i(i(z,z),i(u,v)))),i(u,i(x,v)))),-1).
weight(P(i(i(x,i(y,i(y,z))),i(x,i(y,z)))),-1).
weight(P(i(i(x,i(y,i(z,u))),i(x,i(z,i(y,u))))),-1).
weight(P(i(i(x,i(y,z)),i(i(u,x),i(y,i(u,z))))),-1).
weight(P(i(i(x,i(y,z)),i(i(u,y),i(x,i(u,z))))),-1).
weight(P(i(i(x,i(y,z)),i(x,i(i(z,u),i(y,u))))),-1).
weight(P(i(i(x,y),i(i(i(i(z,u),i(i(u,v),i(z,v))),x),y))),-1).
weight(P(i(i(x,y),i(i(y,z),i(x,z)))),-1).
weight(P(i(x,i(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),w))),-1).
weight(P(i(x,i(i(i(y,y),i(i(z,z),i(x,u))),u))),-1).
weight(P(i(x,i(i(i(y,y),i(x,z)),z))),-1).
weight(P(i(x,i(i(x,y),y))),-1).
weight(P(i(x,i(i(y,i(x,z)),i(y,z)))),-1).
% Following 20 are subformulas of the Rezus for RI, from Beeson.

```
weight(i(i(v6,v7),i(i(v7,v8),i(v6,v8))),1).
weight(i(i(z,z),i(i(u,u),i(i(v,v),i(x,w)))),1).
weight(i(i(v11,v12),i(i(v12,v13),i(v11,v13))),1).
weight(i(i(v19,v19),i(i(v20,v20),i(v16,v21))),1).
weight(i(i(i(v11,v12),i(i(v12,v13),i(v11,v13))),v14),1).
weight(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),1).
weight(i(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),w),1).
weight(i(i(v18,v18),i(i(v19,v19),i(i(v20,v20),i(v16,v21)))),1).
weight(i(x,i(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),w)),1).
weight(i(i(v17,v17),i(i(v18,v18),i(i(v19,v19),i(i(v20,v20),i(v16,v21))))),1).
weight(i(i(i(v17,v17),i(i(v18,v18),i(i(v19,v19),i(i(v20,v20),i(v16,v21))))),v21),1).
weight(i(i(i(v9,i(v9,v10)),i(v9,v10)),i(i(i(v11,v12),i(i(v12,v13),i(v11,v13))),v14)),1).
weight(i(v16,i(i(i(v17,v17),i(i(v18,v18),i(i(v19,v19),i(i(v20,v20),i(v16,v21))))),v21)),1).
weight(i(i(i(i(v9,i(v9,v10)),i(v9,v10)),i(i(i(v11,v12),i(i(v12,v13),i(v11,v13))),v14)),v14),1).
weight(i(i(v16,i(i(i(v17,v17),i(i(v18,v18),i(i(v19,v19),i(i(v20,v20),i(v16,v21))))),v21)),v22),1).
weight(i(i(i(i(i(v9,i(v9,v10)),i(v9,v10)),i(i(i(v11,v12),i(i(v12,v13),i(v11,v13))),v14)),v14),v15),1).
weight(i(i(i(v6,v7),i(i(v7,v8),i(v6,v8))),i(i(i(i(i(v9,i(v9,v10)),i(v9,v10)),i(i(i(v11,v12),i(i(v12,v13),
  i(v11,v13))),v14)),v14),v15)),1).
weight(i(i(i(i(v6,v7),i(i(v7,v8),i(v6,v8))),i(i(i(i(i(v9,i(v9,v10)),i(v9,v10)),i(i(i(v11,v12),
  i(i(v12,v13),i(v11,v13))),v14)),v14),v15)),v15),1).
weight(i(i(i(i(i(v6,v7),i(i(v7,v8),i(v6,v8))),i(i(i(i(i(v9,i(v9,v10)),i(v9,v10)),i(i(i(v11,v12),i(i(v12,v13),
  i(v11,v13))),v14)),v14),v15)),v15),i(i(v16,i(i(i(v17,v17),i(i(v18,v18),i(i(v19,v19),i(i(v20,v20),
  i(v16,v21))))),v21)),v22)),1).
weight(i(i(x,i(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),w)),i(i(i(i(i(v6,v7),i(i(v7,v8),i(v6,v8))),
  i(i(i(i(i(v9,i(v9,v10)),i(v9,v10)),i(i(i(v11,v12),i(i(v12,v13),i(v11,v13))),v14)),v14),v15)),v15),
  i(i(v16,i(i(i(v17,v17),i(i(v18,v18),i(i(v19,v19),i(i(v20,v20),i(v16,v21))))),v21)),v22))),1).
% Following are building blocks for Rezus and Rezus itself.
weight(i(i(x,y),i(i(y,z),i(x,z))),1).   % B'
weight(i(x,i(i(i(y,y),i(x,z)),z)),1).   % Rezus's K1
weight(i(x,i(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),w)),1). % Rezus's K4
weight(P(i(i(i(x,i(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),w)),i(i(i(i(i(v6,v7),i(i(v7,v8),
  i(v6,v8))),i(i(i(i(i(v9,i(v9,v10)),i(v9,v10)),i(i(i(v11,v12),i(i(v12,v13),i(v11,v13))),v14)),v14),v15)),v15),
  i(i(v16,i(i(i(v17,v17),i(i(v18,v18),i(i(v19,v19),i(i(v20,v20),i(v16,v21))))),v21)),v22))),v22)),1).
% A Rezus-style single axiom for RI
end_of_list.

% list(hints).
% % Following 20 derive B C I and W from the new Ulrich 35-symbol single axiom for R.
% P(i(i(u,v),i(i(v,v),i(u,v)))).
% P(i(i(u,i(i(v,w),i(x,v))),i(i(v,w),i(u,i(x,w))))).
% P(i(i(u,i(i(v,v),i(w,v))),i(w,i(u,v)))).
% P(i(i(u,i(i(i(v,w),i(v,w)),i(i(x,x),i(v,w)))),i(v,i(u,w)))).
% P(i(u,i(i(u,v),v))).
% P(i(u,i(i(i(v,v),i(u,w)),w))).
% P(i(i(i(i(u,v),v),i(i(u,v),v)),i(u,i(i(u,v),v)))).
% P(i(i(i(u,i(i(u,v),v)),w),w)).
% P(i(i(i(u,u),i(i(v,i(i(i(w,w),i(v,x)),x)),y)),y)).
% P(i(i(i(i(i(i(u,v),v),i(i(u,v),v)),i(u,i(i(u,v),v))),w),w)).
% P(i(i(i(u,u),i(i(i(v,i(i(v,w),w)),x),x),y)),y)).
% P(i(i(i(u,u),i(i(v,v),w)),w)).
% P(i(u,u)). % I
% P(i(i(u,i(v,w)),i(v,i(u,w)))). % C
% P(i(u,i(i(i(v,v),i(i(i(w,i(i(w,x),x)),y),y),i(u,z))),z))).
```

```
% P(i(i(u,v),i(i(v,w),i(u,w)))). % B'
% P(i(i(u,i(v,w)),i(i(i(v,w),v),i(u,w)))).
% P(i(i(i(i(u,v),i(w,v)),x),i(i(w,u),x))).
% P(i(i(i(i(u,v),v),i(u,v)),i(u,v))).
% P(i(i(u,i(u,v)),i(u,v))). % W
% end_of_list.

list(usable).
-P(i(x,y)) | -P(x) | P(y).
% -P(i(a,a)) | -P(i(i(a,i(b,c)),i(b,i(a,c)))) | -P(i(i(a,b),i(i(b,c),i(a,c)))) | -P(i(i(a,i(a,b)),i(a,b))) | $ANS(all).
end_of_list.

list(sos).
P(i(i(x,y),i(i(y,z),i(x,z)))).   % B'
% P(i(i(u,v),i(i(w,u),i(w,v)))). % B
P(i(i(u,i(v,w)),i(v,i(u,w)))). % C
P(i(u,u)). % I
P(i(i(u,i(u,v)),i(u,v))). % W
end_of_list.

list(passive).
% Following 20 are negs of subformulas of Rezus, from Beeson.
-P(i(i(b6,b7),i(i(b7,b8),i(b6,b8)))) | $ANS(inter4).
-P(i(i(a3,a3),i(i(a4,a4),i(i(b,b),i(a1,a6))))) | $ANS(inter4).
-P(i(i(b11,b12),i(i(b12,b13),i(b11,b13)))) | $ANS(inter4).
-P(i(i(b19,b19),i(i(b20,b20),i(b16,b21)))) | $ANS(inter4).
-P(i(i(i(b11,b12),i(i(b12,b13),i(b11,b13))),b14)) | $ANS(inter4).
-P(i(i(a2,a2),i(i(a3,a3),i(i(a4,a4),i(i(b,b),i(a1,a6)))))) | $ANS(inter4).
-P(i(i(i(a2,a2),i(i(a3,a3),i(i(a4,a4),i(i(b,b),i(a1,a6))))),a6)) | $ANS(inter4).
-P(i(i(b18,b18),i(i(b19,b19),i(i(b20,b20),i(b16,b21))))) | $ANS(inter4).
-P(i(a1,i(i(i(a2,a2),i(i(a3,a3),i(i(a4,a4),i(i(b,b),i(a1,a6))))),a6))) | $ANS(inter4).
-P(i(i(b17,b17),i(i(b18,b18),i(i(b19,b19),i(i(b20,b20),i(b16,b21)))))) | $ANS(inter4).
-P(i(i(i(b17,b17),i(i(b18,b18),i(i(b19,b19),i(i(b20,b20),i(b16,b21))))),b21)) | $ANS(inter4).
-P(i(i(i(b9,i(b9,b10)),i(b9,b10)),i(i(i(b11,b12),i(i(b12,b13),i(b11,b13))),b14))) | $ANS(inter4).
-P(i(b16,i(i(i(b17,b17),i(i(b18,b18),i(i(b19,b19),i(i(b20,b20),i(b16,b21))))),b21))) | $ANS(inter4).
-P(i(i(i(i(b9,i(b9,b10)),i(b9,b10)),i(i(i(b11,b12),i(i(b12,b13),i(b11,b13))),b14)),b14)) | $ANS(inter4).
-P(i(i(b16,i(i(i(b17,b17),i(i(b18,b18),i(i(b19,b19),i(i(b20,b20),i(b16,b21))))),b21)),b22)) | $ANS(inter4).
-P(i(i(i(i(i(b9,i(b9,b10)),i(b9,b10)),i(i(i(b11,b12),i(i(b12,b13),i(b11,b13))),b14)),b14),b15)) | $ANS(inter4).
-P(i(i(i(b6,b7),i(i(b7,b8),i(b6,b8))),i(i(i(i(i(b9,i(b9,b10)),i(b9,b10)),i(i(i(b11,b12),i(i(b12,b13),
  i(b11,b13))),b14)),b14),b15))) | $ANS(inter4).
-P(i(i(i(i(b6,b7),i(i(b7,b8),i(b6,b8))),i(i(i(i(i(b9,i(b9,b10)),i(b9,b10)),i(i(i(b11,b12),i(i(b12,b13),
  i(b11,b13))),b14)),b14),b15)),b15)) | $ANS(inter4).
-P(i(i(i(i(i(b6,b7),i(i(b7,b8),i(b6,b8))),i(i(i(i(i(b9,i(b9,b10)),i(b9,b10)),i(i(i(b11,b12),i(i(b12,b13),
  i(b11,b13))),b14)),b14),b15)),b15),i(i(b16,i(i(i(b17,b17),i(i(b18,b18),i(i(b19,b19),i(i(b20,b20),
  i(b16,b21))))),b21)),b22))) | $ANS(inter4).
-P(i(i(a1,i(i(i(a2,a2),i(i(a3,a3),i(i(a4,a4),i(i(b,b),i(a1,a6))))),a6)),i(i(i(i(b6,b7),i(i(b7,b8),i(b6,b8))),
  i(i(i(i(i(b9,i(b9,b10)),i(b9,b10)),i(i(i(b11,b12),i(i(b12,b13),i(b11,b13))),b14)),b14),b15)),b15),
  i(i(b16,i(i(i(b17,b17),i(i(b18,b18),i(i(b19,b19),i(i(b20,b20),i(b16,b21))))),b21)),b22)))) | $ANS(inter4).
% Following are negs of B', K1, K4, and Rezus, the first 3 are building blocks for Rezus.
-P(i(i(a1,a2),i(i(a2,a3),i(a1,a3)))) | $ANS(BP).
-P(i(a1,i(i(i(a2,a2),i(a1,a3)),a3))) | $ANS(K1).
-P(i(a1,i(i(i(a2,a2),i(i(a3,a3),i(i(a4,a4),i(i(a5,a5),i(a1,a6))))),a6))) | $ANS(K4).
-P(i(i(i(a1,i(i(i(a2,a2),i(i(a3,a3),i(i(a4,a4),i(i(b,b),i(a1,a6))))),a6)),i(i(i(i(b6,b7),i(i(b7,b8),
```

```
  i(b6,b8))),i(i(i(i(i(b9,i(b9,b10)),i(b9,b10)),i(i(i(b11,b12),i(i(b12,b13),i(b11,b13))),b14)),b14),b15)),b15),
  i(i(b16,i(i(i(b17,b17),i(i(b18,b18),i(i(b19,b19),i(i(b20,b20),i(b16,b21))))),b21)),b22))),b22)) | $ANS(REZ).
end_of_list.

list(demodulators).
%  (i(i(x,x),y) = junk).
%  (i(y,i(x,x)) = junk).
(i(x,junk) = junk).
(i(junk,x) = junk).
(P(junk) = $T).
end_of_list.
```

As for the other side of the coin, the shortest proof I have in hand showing that the Rezus formula implies the 4-basis featured here has length 15 and level 10.

A 15-Step Proof Showing that Rezus Implies a 4-Basis

```
----- Otter 3.3g-work, Jan 2005 -----
The process was started by wos on lemma.mcs.anl.gov,
Sun Apr 13 17:27:55 2008
The command was "otter".  The process ID is 3687.

-----> EMPTY CLAUSE at   1.11 sec ----> 3168 [hyper,67,98,82,507,101] $ANS(allBPCIW).

Length of proof is 15.  Level of proof is 10.

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

66 [] -P(i(x,y)) | -P(x) |P(y).
67 [] -P(i(a,a)) | -P(i(i(a,b),i(i(b,c),i(a,c)))) | -P(i(i(a,i(b,c)),i(b,i(a,c)))) |
  -P(i(i(a,i(a,b)),i(a,b))) |$ANS(allBPCIW).
68 [] P(i(i(i(x,i(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),w)),i(i(i(i(v6,v7),i(i(v7,v8),i(v6,v8))),
  i(i(i(i(v9,i(v9,v10)),i(v9,v10)),i(i(i(v11,v12),i(i(v12,v13),i(v11,v13))),v14)),v14),v15)),v15),
  i(i(v16,i(i(i(v17,v17),i(i(v18,v18),i(i(v19,v19),i(i(v20,v20),i(v16,v21))))),v21)),v22))),v22)).
78 [hyper,66,68,68] P(i(i(i(x,x),i(i(y,y),i(i(z,z),i(i(u,u),i(v,i(i(i(w,w),i(i(v6,v6),i(i(v7,v7),
  i(i(v8,v8),i(v,v9))))),v9)),v10))))),v10)).
79 [hyper,66,78,78] P(i(x,i(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),w))).
80 [hyper,66,68,78] P(i(i(i(i(x,y),i(i(y,z),i(x,z))),i(i(i(i(u,i(u,v)),i(u,v)),i(i(i(w,v6),
  i(i(v6,v7),i(w,v7))),v8)),v8),v9)),v9)).
82 [hyper,66,80,79] P(i(i(x,y),i(i(y,z),i(x,z)))).
86 [hyper,66,82,82] P(i(i(i(i(x,y),i(z,y)),u),i(i(z,x),u))).
90 [hyper,66,82,79] P(i(i(i(i(i(x,x),i(i(y,y),i(i(z,z),i(i(u,u),i(v,w))))),w),v6),i(v,v6))).
94 [hyper,66,90,80] P(i(i(i(i(i(x,i(x,y)),i(x,y)),i(i(i(z,u),i(i(u,v),i(z,v))),w)),w),v6),v6)).
98 [hyper,66,90,94] P(i(x,x)).
101 [hyper,66,68,94] P(i(i(x,i(x,y)),i(x,y))).
110 [hyper,66,86,82] P(i(i(x,y),i(i(i(x,z),u),i(i(y,z),u)))).
124 [hyper,66,86,110] P(i(i(x,y),i(i(i(i(y,z),u),v),i(i(i(x,z),u),v)))).
143 [hyper,66,86,124] P(i(i(x,y),i(i(i(i(i(x,z),u),v),w),i(i(i(i(y,z),u),v),w)))).
189 [hyper,66,124,143] P(i(i(i(i(i(i(i(i(i(x,y),z),u),v),i(i(i(i(w,y),z),u),v)),v6),v7),v8),
  i(i(i(i(x,w),v6),v7),v8))).
216 [hyper,66,189,90] P(i(i(i(i(x,y),y),z),i(x,z))).
507 [hyper,66,86,216] P(i(i(x,i(y,z)),i(y,i(x,z)))).
```

The approach I used to obtain the just-given proof, as you can see from the input file I shall supply, is based in part on cramming, in part on demodulation blocking, and in part on the resonance strategy. The approach is iterative, as it is with almost all of my studies aimed at finding a shorter proof.

An Input File That Yields a 15-Step Proof Showing Rezus Implies a 4-Basis

```
set(hyper_res).
assign(max_mem,880000).
% assign(max_seconds,6).
% set(sos_queue).
assign(max_weight,40).
% assign(change_limit_after,1500).
% assign(new_max_weight,18).
assign(max_proofs,-1).
assign(pick_given_ratio,1).
assign(bsub_hint_wt,1).
set(keep_hint_subsumers).
% set(keep_hint_equivalents).
set(ancestor_subsume).
set(back_sub).
set(order_history).
% set(process_input).
clear(print_kept).

weight_list(pick_and_purge).
% Following 20/10?, temp.ulrich.singax.r.out3o, avoids 8 of the 17.
weight(P(i(i(i(x,x),i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,i(i(i(w,w),i(i(v6,v6),i(i(v7,v7),i(i(v8,v8),
  i(v,v9))))),v9)),v10))))),v10)),-6).
weight(P(i(x,i(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),w))),-6).
weight(P(i(i(i(i(x,y),i(i(y,z),i(x,z))),i(i(i(i(u,i(u,v)),i(u,v)),i(i(w,v6),i(i(v6,v7),
  i(w,v7))),v8)),v8),v9)),v9)),-6).
weight(P(i(i(x,y),i(i(y,z),i(x,z)))),-6).
weight(P(i(i(i(i(x,i(x,y)),i(x,y)),i(i(i(z,u),i(i(u,v),i(z,v))),w)),w)),-6).
weight(P(i(i(i(i(x,y),i(z,y)),u),i(i(z,x),u))),-6).
weight(P(i(i(i(i(i(x,x),i(i(y,y),i(i(z,z),i(i(u,u),i(v,w))))),w),v6),i(v,v6))),-6).
weight(P(i(i(x,i(x,y)),i(x,y))),-6).
weight(P(i(i(x,y),i(i(i(x,z),u),i(i(y,z),u)))),-6).
weight(P(i(i(x,y),i(i(i(i(y,z),u),v),i(i(i(x,z),u),v)))),-6).
weight(P(i(i(i(i(i(x,y),z),i(i(u,y),z)),v),i(i(x,u),v))),-6).
weight(P(i(i(x,y),i(i(i(i(i(x,z),u),v),w),i(i(i(i(y,z),u),v),w)))),-6).
weight(P(i(i(x,y),i(i(i(i(i(i(y,z),u),v),w),v6),i(i(i(i(i(x,z),u),v),w),v6)))),-6).
weight(P(i(i(x,i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(w,v6)))))),i(w,i(x,v6)))),-6).
weight(P(i(i(i(i(i(i(x,i(x,y)),i(x,y)),i(i(i(z,u),i(i(u,v),i(z,v))),w)),w),v6),v6)),-6).
weight(P(i(i(x,y),i(i(z,x),i(z,y)))),-6).
weight(P(i(i(i(i(i(i(i(i(i(x,y),z),u),v),i(i(i(i(w,y),z),u),v)),v6),v7),v8),i(i(i(x,w),v6),v7),v8))),-6).
weight(P(i(x,x)),-6).
weight(P(i(i(i(i(x,y),y),z),i(x,z))),-6).
weight(P(i(i(x,i(y,z)),i(y,i(x,z)))),-6).
% Following 6 prove B, based some or all of the first 12 of a 17-step proof that proves BCIW
  from Rezus; with 5 not in the 17.
weight(P(i(i(x,y),i(i(i(x,z),u),i(i(y,z),u)))),-4).
weight(P(i(i(x,i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(w,v6)))))),i(w,i(x,v6)))),-4).
weight(P(i(i(x,y),i(i(i(i(y,z),u),v),i(i(i(x,z),u),v)))),-4).
```

```
weight(P(i(i(i(i(i(x,y),z),i(i(u,y),z)),v),i(i(x,u),v))),-4).
weight(P(i(i(x,y),i(i(i(i(i(y,z),u),v),w),v6),i(i(i(i(x,z),u),v),w),v6)))),-4).
weight(P(i(i(x,y),i(i(z,x),i(z,y)))),-4).
end_of_list.

list(hints).
% Following 16/12 prove from Rezus B', C, I, and w.
P(i(i(i(x,x),i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,i(i(i(w,w),i(i(v6,v6),i(i(v7,v7),i(i(v8,v8),i(v,v9))))),v9)),v10))))),v10)).
P(i(x,i(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),w))).
P(i(i(i(i(x,y),i(i(y,z),i(x,z))),i(i(i(i(i(u,i(u,v)),i(u,v)),i(i(i(w,v6),i(i(v6,v7),i(w,v7))),v8)),v8),v9)),v9)).
P(i(i(x,y),i(i(y,z),i(x,z)))).
P(i(i(i(i(x,y),i(z,y)),u),i(i(z,x),u))).
P(i(i(i(i(x,y),i(i(y,z),i(x,z))),i(u,i(u,v))),i(u,v))).
P(i(i(x,i(y,z)),i(i(u,y),i(x,i(u,z))))).
P(i(i(x,y),i(i(z,x),i(i(y,u),i(z,u))))).
P(i(i(x,i(i(i(y,z),i(u,z)),v)),i(i(i(i(u,y),v),w),i(x,w)))).
P(i(i(i(i(i(x,x),i(i(y,y),i(i(z,z),i(u,v)))),v),w),i(u,w))).
P(i(x,i(i(i(y,y),i(i(z,z),i(x,u))),u))).
P(i(i(x,i(x,y)),i(x,y))).
P(i(i(x,i(i(y,y),i(i(z,z),i(u,v)))),i(u,i(x,v)))).
P(i(x,x)).
P(i(x,i(i(x,y),y))).
P(i(i(x,i(y,z)),i(y,i(x,z)))).
% Following 19 prove from Rezus B C W I.
P(i(i(i(x,x),i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,i(i(i(w,w),i(i(v6,v6),i(i(v7,v7),i(i(v8,v8),i(v,v9))))),v9)),v10))))),v10)).
P(i(x,i(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),w))).
P(i(i(i(i(x,y),i(i(y,z),i(x,z))),i(i(i(i(i(u,i(u,v)),i(u,v)),i(i(i(w,v6),i(i(v6,v7),i(w,v7))),v8)),v8),v9)),v9)).
P(i(i(x,y),i(i(y,z),i(x,z)))).
P(i(i(i(i(x,y),i(z,y)),u),i(i(z,x),u))).
P(i(i(i(i(x,y),i(i(y,z),i(x,z))),i(u,i(u,v))),i(u,v))).
P(i(i(x,i(y,z)),i(i(u,y),i(x,i(u,z))))).
P(i(i(i(i(x,y),z),u),i(i(i(i(y,v),i(x,v)),z),u))).
P(i(i(x,i(y,z)),i(i(i(i(z,u),i(y,u)),v),i(x,v)))).
P(i(i(x,i(i(i(y,z),i(u,z)),v)),i(i(i(i(u,y),v),w),i(x,w)))).
P(i(i(x,i(i(i(y,z),i(u,z)),v)),i(i(w,i(u,y)),i(x,i(w,v))))).
P(i(i(i(i(i(x,x),i(i(y,y),i(i(z,z),i(u,v)))),v),w),i(u,w))).
P(i(i(x,i(i(y,y),i(i(z,z),i(i(u,u),i(v,w))))),i(v,i(x,w)))).
P(i(x,i(i(i(y,y),i(i(z,z),i(x,u))),u))).
P(i(i(x,i(x,y)),i(x,y))).
P(i(x,i(i(i(y,y),i(x,z)),z))).
P(i(i(x,i(i(y,y),i(i(z,z),i(u,v)))),i(u,i(x,v)))).
P(i(i(x,i(y,z)),i(y,i(x,z)))).
P(i(x,x)).
% Following 17 derive B C I and W from the new Ulrich 35-symbol single axiom for R.
P(i(i(i(i(i(i(x,y),y),i(i(x,y),y)),i(x,i(i(x,y),y))),z),z)).
P(i(i(i(i(x,y),i(z,y)),u),i(i(z,x),u))).
P(i(i(i(i(x,y),y),i(i(x,y),y)),i(x,i(i(x,y),y)))).
P(i(i(i(i(x,y),y),i(x,y)),i(x,y))).
P(i(i(i(i(x,y),y),z),i(x,z))).
P(i(i(i(x,i(i(x,y),y)),z),i(i(i(i(x,y),y),i(i(x,y),y)),z))).
P(i(i(x,i(i(y,y),i(z,y))),i(z,i(x,y)))).
P(i(i(x,i(i(y,y),y)),i(x,y))).
P(i(i(x,i(i(y,z),i(u,y))),i(i(y,z),i(x,i(u,z))))).
```

```
P(i(i(x,i(x,y)),i(x,y))).
P(i(i(x,i(y,z)),i(i(i(y,z),y),i(x,z)))).
P(i(i(x,i(y,z)),i(y,i(x,z)))).
P(i(i(x,y),i(i(y,y),i(x,y)))).
P(i(i(x,y),i(i(y,z),i(x,z)))).
P(i(x,i(i(x,y),y))).
P(i(x,i(y,i(i(x,i(y,z)),z)))).
P(i(x,x)).
%  Following 13/5 prove Ulrich-35 from B C I W.
P(i(x,i(i(y,i(x,z)),i(y,z)))).
P(i(i(x,y),i(i(z,x),i(z,y)))).
P(i(i(x,i(i(y,y),z)),i(x,z))).
P(i(i(i(i(x,y),i(x,z)),u),i(i(y,z),u))).
P(i(i(x,i(y,i(y,z))),i(x,i(y,z)))).
P(i(i(x,i(y,i(z,u))),i(x,i(z,i(y,u))))).
P(i(i(x,i(y,z)),i(x,i(i(z,u),i(y,u))))).
P(i(x,i(i(x,i(i(y,y),z)),z))).
P(i(i(i(x,y),i(z,x)),i(i(x,y),i(z,y)))).
P(i(i(i(i(i(x,y),i(z,x)),i(i(x,y),i(z,y))),i(i(u,u),v)),v)).
P(i(i(x,i(y,z)),i(i(u,x),i(y,i(u,z))))).
P(i(i(x,i(y,i(z,u))),i(x,i(i(v,y),i(z,i(v,u)))))).
P(i(i(i(i(i(x,y),i(z,x)),i(i(x,y),i(z,y))),i(i(u,u),i(v,i(w,v6)))),i(i(v7,v),i(w,i(v7,v6))))).
end_of_list.

list(usable).
-P(i(x,y)) | -P(x) | P(y).
-P(i(a,a)) | -P(i(i(a,b),i(i(b,c),i(a,c)))) | -P(i(i(a,i(b,c)),i(b,i(a,c)))) | -P(i(i(a,i(a,b)),i(a,b))) | $ANS(allBPCIW).
end_of_list.

list(sos).
%  Following is the Rezus single axiom for RI; preceding is Ulrich's single.
P(i(i(i(x,i(i(i(y,y),i(i(z,z),i(i(u,u),i(i(v,v),i(x,w))))),w)),i(i(i(i(v6,v7),i(i(v7,v8),i(v6,v8))),
  i(i(i(i(i(v9,i(v9,v10)),i(v9,v10)),i(i(i(v11,v12),i(i(v12,v13),i(v11,v13))),v14)),v14),v15)),v15),
  i(i(v16,i(i(i(v17,v17),i(i(v18,v18),i(i(v19,v19),i(i(v20,v20),i(v16,v21))))),v21)),v22))),v22)).
% A Rezus-style single axiom for RI
end_of_list.

list(passive).
-P(i(a,a)) | $ANS(I).
-P(i(i(a,i(b,c)),i(b,i(a,c)))) | $ANS(C).
-P(i(i(a,b),i(i(b,c),i(a,c)))) | $ANS(BP).
%  -P(i(i(a,b),i(i(c,a),i(c,b)))) | $ANS(B).
-P(i(i(a,i(a,b)),i(a,b))) | $ANS(W).
end_of_list.

list(demodulators).
(P(i(i(i(i(x,i(x,y)),i(x,y)),i(i(i(z,u),i(i(u,v),i(z,v))),w)),w)) = junk).
(P(i(i(x,y),i(i(i(i(i(y,z),u),v),w),v6),i(i(i(i(x,z),u),v),w),v6)))) = junk).
%  (P(i(i(i(i(x,y),z),u),i(i(i(i(y,v),i(x,v)),z),u))) = junk).
%  (P(i(x,i(i(i(y,y),i(x,z)),z))) = junk).
%  (P(i(i(x,i(y,z)),i(i(i(z,u),i(y,u)),v),i(x,v)))) = junk).
%  (i(i(x,x),y) = junk).
%  (i(y,i(x,x)) = junk).
```

```
(i(x,junk) = junk).
(i(junk,x) = junk).
(P(junk) = $T).
end_of_list.
```

As expected, with the presentation of various proofs, I implicitly offer challenges for the individual who might find intriguing the pursuit of ever-shorter proofs. For a quite different challenge, you may have noticed that I did not offer results in the context of using as hypothesis the Ulrich 35-symbol single axiom and as target the 4-basis consisting of *B*', *C*, *I*, and *W*, where the subformula strategy is in play along with (possibly) resonators corresponding to the four given targets. The idea in this challenge is to locate some (if they exist) parameter values and some settings of the options to enable a program (perhaps OTTER) to find the desired proof. The assignment may be far from trivial. If you wish to try your hand at meeting the challenge with no guidance or comment, then be warned, for I shall attempt to meet it, giving my results quite late in this notebook.

At this point, not too far from the close of this notebook, I return to some results focusing on the Ulrich 35-symbol single axiom. Among these you will find one particular success that surprised and delighted me, a dividend resulting from a study not designed to unearth the treasure that was found. The first item concerns the shortest proof known to me that derives the (now famous here) 4-basis from the Ulrich single axiom.

A 17-Step Proof of the Deducibility of the 4-Basis from the Ulrich Single Axiom

```
----- Otter 3.3g-work, Jan 2005 -----
The process was started by wos on elephant.mcs.anl.gov,
Thu Oct 11 16:53:09 2007
The command was "otter". The process ID is 11438.

-----> EMPTY CLAUSE at   0.68 sec ----> 3657 [hyper,22,983,150,79,3563] $ANS(all).

Length of proof is 17. Level of proof is 12.

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

21 [] -P(i(x,y)) | -P(x) | P(y).
22 [] -P(i(a,a)) | -P(i(i(a,i(b,c)),i(b,i(a,c)))) | -P(i(i(a,b),i(i(b,c),i(a,c)))) | -P(i(i(a,i(a,b)),i(a,b))) | $ANS(all).
23 [] P(i(i(i(i(i(u,v),i(w,u)),i(i(u,v),i(w,v))),i(i(x,x),i(y,i(z,v6)))),i(i(v7,y),i(z,i(v7,v6))))).
33 [hyper,21,23,23] P(i(i(x,y),i(i(y,y),i(x,y)))).
35 [hyper,21,23,33] P(i(i(x,i(i(y,z),i(u,y))),i(i(y,z),i(x,i(u,z))))).
41 [hyper,21,23,35] P(i(i(x,i(i(y,y),i(z,y))),i(z,i(x,y)))).
47 [hyper,21,41,33] P(i(x,i(i(x,y),y))).
54 [hyper,21,33,47] P(i(i(i(i(x,y),y),i(i(x,y),y)),i(x,i(i(x,y),y)))).
68 [hyper,21,47,54] P(i(i(i(i(i(i(x,y),y),i(i(x,y),y)),i(x,i(i(x,y),y))),z),z)).
79 [hyper,21,23,68] P(i(i(x,y),i(i(y,z),i(x,z)))).
85 [hyper,21,79,79] P(i(i(i(i(x,y),i(z,y)),u),i(i(z,x),u))).
90 [hyper,21,79,47] P(i(i(i(i(x,y),y),z),i(x,z))).
131 [hyper,21,85,23] P(i(i(i(x,x),i(i(y,z),i(u,y))),i(i(v,i(y,z)),i(u,i(v,z))))).
150 [hyper,21,85,90] P(i(i(x,i(y,z)),i(y,i(x,z)))).
177 [hyper,21,90,150] P(i(x,i(y,i(i(x,i(y,z)),z)))).
395 [hyper,21,41,177] P(i(i(x,i(i(y,y),y)),i(x,y))).
983 [hyper,21,395,47] P(i(x,x)).
3331 [hyper,21,131,90] P(i(i(x,i(y,z)),i(i(i(y,z),y),i(x,z)))).
3443 [hyper,21,3331,47] P(i(i(i(i(x,y),y),i(x,y)),i(x,y))).
3563 [hyper,21,85,3443] P(i(i(x,i(x,y)),i(x,y))).
```

The approach I used to find the given proof was based mainly on demodulation blocking, as evidenced in the following input file in which two formulas are demodulated to junk and on the use of Veroff's *hints strategy*. *Hints*, like resonators, are used to direct a program's reasoning; they themselves do not take on a **true** or **false** value. Different from resonators, clauses that subsume or are subsumed by a hint are given weight (priority) based on items in the input list, whereas resonators focus on the functional pattern, treating all variables as indistinguishable.

An Input File That Led to Finding a 17-Step Proof Deriving a 4-Basis from the Ulrich Single Axiom

```
set(hyper_res).
assign(max_mem,880000).
% assign(max_seconds,2).
% 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,1).
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(hints).
% Following 20 derive B C I and W from the new Ulrich 35-symbol single axiom for R.
P(i(i(u,v),i(i(v,v),i(u,v)))).
P(i(i(u,i(i(v,w),i(x,v))),i(i(v,w),i(u,i(x,w))))).
P(i(i(u,i(i(v,v),i(w,v))),i(w,i(u,v)))).
P(i(i(u,i(i(i(v,w),i(v,w)),i(i(x,x),i(v,w)))),i(v,i(u,w)))).
P(i(u,i(i(u,v),v))).
P(i(u,i(i(i(v,v),i(u,w)),w))).
P(i(i(i(i(u,v),v),i(i(u,v),v)),i(u,i(i(u,v),v)))).
P(i(i(i(u,i(i(u,v),v)),w),w)).
P(i(i(i(u,u),i(i(v,i(i(i(w,w),i(v,x)),x)),y)),y)).
P(i(i(i(i(i(i(u,v),v),i(i(u,v),v)),i(u,i(i(u,v),v))),w),w)).
P(i(i(i(u,u),i(i(i(i(v,i(i(v,w),w)),x),x),y)),y)).
P(i(i(i(u,u),i(i(v,v),w)),w)).
P(i(u,u)). % I
P(i(i(u,i(v,w)),i(v,i(u,w)))). % C
P(i(u,i(i(i(v,v),i(i(i(i(w,i(i(w,x),x)),y),y),i(u,z))),z))).
P(i(i(u,v),i(i(v,w),i(u,w)))). % B’
P(i(i(u,i(v,w)),i(i(i(v,w),v),i(u,w)))).
P(i(i(i(i(u,v),i(w,v)),x),i(i(w,u),x))).
P(i(i(i(i(u,v),v),i(u,v)),i(u,v))).
P(i(i(u,i(u,v)),i(u,v))). % W
end_of_list.

list(usable).
-P(i(x,y)) | -P(x) | P(y).
-P(i(a,a)) | -P(i(i(a,i(b,c)),i(b,i(a,c)))) | -P(i(i(a,b),i(i(b,c),i(a,c)))) | -P(i(i(a,i(a,b)),i(a,b))) | $ANS(all).
end_of_list.

list(sos).
P(i(i(i(i(i(u,v),i(w,u)),i(i(u,v),i(w,v))),i(i(x,x),i(y,i(z,v6)))),i(i(v7,y),i(z,i(v7,v6))))).
end_of_list.

list(passive).
```

```
-P(i(a,a)) | $ANS(I).
-P(i(i(a,i(b,c)),i(b,i(a,c)))) | $ANS(C).
-P(i(i(a,b),i(i(b,c),i(a,c)))) | $ANS(B’).
-P(i(i(a,i(a,b)),i(a,b))) | $ANS(W).
end_of_list.

list(demodulators).
(P(i(i(i(x,i(i(x,y),y)),z),z)) = junk).
(P(i(i(x,i(i(i(y,z),i(y,z)),i(i(u,u),i(y,z)))),i(y,i(x,z)))) = junk).
% (i(i(x,x),y) = junk).
% (i(y,i(x,x)) = junk).
(i(x,junk) = junk).
(i(junk,x) = junk).
(P(junk) = $T).
end_of_list.
```

Sometimes, but of course not often, a study of one aspect of a field leads to the unexpected—to the answering of a question (focusing on another aspect) that has proved to be difficult to answer. When that occurs, at least for me, utter joy is the result. As is so dominant in this notebook, I have spent many, many days—even weeks—exploring the power of the new subformula strategy. On the whole, I am satisfied. In an earlier notebook, titled “The Subformula Strategy: Coping with Complex Expressions”, I studied the subformula strategy in various contexts. When I sought to prove diverse theorems with the strategy, I relied also on “building blocks” supplied to me by a colleague. In this notebook, I took a so-called purer approach, omitting the use of building blocks. The successes have been presented earlier here.

And now for a discussion of the “unexpected”. While developing the material offered in the same earlier notebook, I found, in the logic *Ri*, two 13-step proofs establishing the deducibility of the Ulrich 35-symbol single axiom. In both cases, the hypothesis was the 4-basis featured here, namely, *B*’, *C*, *I*, and *W*. With those two proofs in hand, I tried repeatedly, to no avail, to find a path to a proof strictly shorter than length 13. My interest and, yes, perhaps hope, was rekindled by the discovery of a 17-step level-5 proof that deduces, from the cited 4-basis, the Ulrich single axiom. If memory serves, the rekindling resulted from an examination of the 17-step proof, an examination that showed this proof to be sharply different from either 13-step proof already in hand. My renewed effort paid off when I used the following input file.

An Input File Yielding a Breakthrough in Proof Shortening

```
set(hyper_res).
assign(max_mem,880000).
% assign(max_seconds,3).
% set(sos_queue).
assign(max_weight,8).
% assign(change_limit_after,1100).
% assign(new_max_weight,8).
assign(max_proofs,-1).
assign(pick_given_ratio,1).
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).
```

```
weight_list(pick_and_purge).
% Following 13/8, temp.ulrich.singax.r.out2w8l, are quite unlike most of the 13's and 14's.
weight(P(i(i(x,y),i(i(z,x),i(z,y)))),1).
weight(P(i(i(x,i(y,z)),i(x,i(i(u,y),i(u,z))))),1).
weight(P(i(i(x,i(y,i(y,z))),i(x,i(y,z)))),1).
weight(P(i(i(x,i(y,i(z,u))),i(x,i(z,i(y,u))))),1).
weight(P(i(i(x,i(y,z)),i(x,i(i(z,u),i(y,u))))),1).
weight(P(i(x,i(i(x,i(y,z)),i(i(u,y),i(u,z))))),1).
weight(P(i(i(x,y),i(z,i(i(z,x),y)))),1).
weight(P(i(i(i(x,y),i(z,x)),i(i(x,y),i(z,y)))),1).
weight(P(i(i(i(x,x),i(y,z)),i(i(u,y),i(u,z)))),1).
weight(P(i(i(i(i(x,y),i(x,z)),u),i(i(i(v,v),i(y,z)),u))),1).
weight(P(i(i(i(x,x),i(y,i(z,u))),i(i(v,y),i(z,i(v,u))))),1).
weight(P(i(x,i(i(x,i(i(y,y),i(z,i(u,v)))),i(i(w,z),i(u,i(w,v)))))),1).
weight(P(i(i(i(i(i(x,y),i(z,x)),i(i(x,y),i(z,y))),i(i(u,u),i(v,i(w,v6)))),i(i(v7,v),i(w,i(v7,v6))))),1).
% Following 14/7 from temp.ulrich.singax.r.out2w8k is quite different when compared with an earlier
  14/6 and the 13/5 and 13/6.
weight(P(i(x,i(i(x,y),y))),1).
weight(P(i(i(x,y),i(i(z,x),i(z,y)))),1).
weight(P(i(i(x,i(y,z)),i(x,i(i(u,y),i(u,z))))),1).
weight(P(i(i(x,i(y,i(y,z))),i(x,i(y,z)))),1).
weight(P(i(i(x,i(y,i(z,u))),i(x,i(z,i(y,u))))),1).
weight(P(i(i(x,i(y,z)),i(x,i(i(z,u),i(y,u))))),1).
weight(P(i(x,i(i(x,i(y,z)),i(i(u,y),i(u,z))))),1).
weight(P(i(i(i(x,y),i(z,x)),i(i(x,y),i(z,y)))),1).
weight(P(i(i(i(i(i(x,y),i(z,x)),i(i(x,y),i(z,y))),u),u)),1).
weight(P(i(i(i(x,x),i(y,z)),i(i(u,y),i(u,z)))),1).
weight(P(i(i(i(i(x,y),i(x,z)),u),i(i(i(v,v),i(y,z)),u))),1).
weight(P(i(i(x,y),i(i(i(i(i(z,u),i(v,z)),i(i(z,u),i(v,u))),x),y))),1).
weight(P(i(i(i(x,x),i(y,i(z,u))),i(i(v,y),i(z,i(v,u))))),1).
weight(P(i(i(i(i(i(x,y),i(z,x)),i(i(x,y),i(z,y))),i(i(u,u),i(v,i(w,v6)))),i(i(v7,v),i(w,i(v7,v6))))),1).
weight(i(i(i(i(i(u,v),i(w,u)),i(i(u,v),i(w,v))),i(i(x,x),i(y,i(z,v6)))),i(i(v7,y),i(z,i(v7,v6)))),1).
% Ulrich 35-symbol formula
% Following 8 from Beeson are subformulas of the Ulrich formula.
weight(i(v7,v6),1).
weight(i(i(v7,y),i(z,i(v7,v6))),1).
weight(i(y,i(z,v6)),1).
weight(i(i(x,x),i(y,i(z,v6))),1).
weight(i(i(u,v),i(w,v)),1).
weight(i(i(i(u,v),i(w,u)),i(i(u,v),i(w,v))),1).
weight(i(i(i(i(u,v),i(w,u)),i(i(u,v),i(w,v))),i(i(x,x),i(y,i(z,v6)))),1).
weight(i(i(i(i(i(u,v),i(w,u)),i(i(u,v),i(w,v))),i(i(x,x),i(y,i(z,v6)))),i(i(v7,y),i(z,i(v7,v6)))),1).
end_of_list.

list(usable).
-P(i(x,y)) | -P(x) | P(y).
end_of_list.

list(sos).
P(i(i(u,v),i(i(v,w),i(u,w)))). % B'
P(i(i(u,i(v,w)),i(v,i(u,w)))). % C
P(i(u,u)). % I
P(i(i(u,i(u,v)),i(u,v))). % W
```

```
end_of_list.

list(passive).
-P(i(i(i(i(i(a1,a2),i(a3,a1)),i(i(a1,a2),i(a3,a2))),i(i(a4,a4),i(a5,i(a6,a7)))),i(i(a8,a5),
  i(a6,i(a8,a7))))) | $ANS(U35).
% Following 3 are negs of subformulas of the Ulrich.
-P(i(i(i(a1,b),i(a3,a1)),i(i(a1,b),i(a3,b)))) | $ANS(inter).
-P(i(i(a4,a4),i(i(a5,i(a6,b6)),i(i(b7,a5),i(a6,i(b7,b6)))))) | $ANS(inter).
-P(i(i(a5,i(a6,b6)),i(i(b7,a5),i(a6,i(b7,b6))))) | $ANS(inter).
-P(i(a,a)) | $ANS(I).
-P(i(i(a,i(b,c)),i(b,i(a,c)))) | $ANS(C).
-P(i(i(a,b),i(i(b,c),i(a,c)))) | $ANS(B').
-P(i(i(a,i(a,b)),i(a,b))) | $ANS(W).
end_of_list.

list(demodulators).
(P(i(i(x,i(y,z)),i(x,i(i(u,y),i(u,z))))) = junk).
% (P(i(i(i(x,i(i(x,y),y)),z),z)) = junk).
% (P(i(i(x,i(i(i(y,z),i(y,z)),i(i(u,u),i(y,z)))),i(y,i(x,z)))) = junk).
% (i(i(x,x),y) = junk).
% (i(y,i(x,x)) = junk).
(i(x,junk) = junk).
(i(junk,x) = junk).
(P(junk) = $T).
end_of_list.
```

The following was returned to me by OTTER.

A New and Shorter Proof of Interest

```
----- Otter 3.3g-work, Jan 2005 -----
The process was started by wos on elephant.mcs.anl.gov,
Thu May  1 17:29:47 2008
The command was "otter".  The process ID is 21825.

----> UNIT CONFLICT at   1.54 sec ----> 3867 [binary,3866.1,6.1] $ANS(U35).

Length of proof is 12.  Level of proof is 9.

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

1 [] -P(i(x,y))| -P(x)|P(y).
2 [] P(i(i(u,v),i(i(v,w),i(u,w)))).
3 [] P(i(i(u,i(v,w)),i(v,i(u,w)))).
4 [] P(i(u,u)).
5 [] P(i(i(u,i(u,v)),i(u,v))).
6 [] -P(i(i(i(i(i(a1,a2),i(a3,a1)),i(i(a1,a2),i(a3,a2))),i(i(a4,a4),i(a5,i(a6,a7)))),i(i(a8,a5),
  i(a6,i(a8,a7)))))|$ANS(U35).
22 [hyper,1,3,2] P(i(i(x,y),i(i(z,x),i(z,y)))).
34 [hyper,1,22,5] P(i(i(x,i(y,i(y,z))),i(x,i(y,z)))).
35 [hyper,1,22,3] P(i(i(x,i(y,i(z,u))),i(x,i(z,i(y,u))))).
36 [hyper,1,22,2] P(i(i(x,i(y,z)),i(x,i(i(z,u),i(y,u))))).
60 [hyper,1,35,22] P(i(i(x,y),i(z,i(i(z,x),y)))).
```

```
70 [hyper,1,34,36] P(i(i(i(x,y),i(z,x)),i(i(x,y),i(z,y)))).
148 [hyper,1,60,22] P(i(x,i(i(x,i(y,z)),i(i(u,y),i(u,z))))).
274 [hyper,1,148,4] P(i(i(i(x,x),i(y,z)),i(i(u,y),i(u,z)))).
509 [hyper,1,2,274] P(i(i(i(i(x,y),i(x,z)),u),i(i(i(v,v),i(y,z)),u))).
981 [hyper,1,509,35] P(i(i(i(x,x),i(y,i(z,u))),i(i(v,y),i(z,i(v,u))))).
1735 [hyper,1,60,981] P(i(x,i(i(x,i(i(y,y),i(z,i(u,v)))),i(i(w,z),i(u,i(w,v)))))).
3866 [hyper,1,1735,70] P(i(i(i(i(i(x,y),i(z,x)),i(i(x,y),i(z,y))),i(i(u,u),i(v,i(w,v6)))),
  i(i(v7,v),i(w,i(v7,v6))))).
```

Some person not as pleased as I was about the new discovery might focus on the small reduction in proof length, namely, a single step. However, when a comparison is made of the new 12-step proof with either of the 13-step proofs, perhaps more excitement will occur, enough to be startling. Indeed, the new 12-step proof contains six steps (formulas) not in either 13-step proof. In that the genesis of this study that produced the 12-step proof was a run relying solely on eight subformulas, I must report a sense of substantial piquancy.

Because some readers would enjoy comparing in detail and on their own the newer proof with one of the 13-step proofs, I now include what is needed.

A 13-Step Proof for Comparison

```
----- Otter 3.3g-work, Jan 2005 -----
The process was started by wos on elephant.mcs.anl.gov,
Fri Oct 19 16:56:28 2007
The command was "otter". The process ID is 3026.

----> UNIT CONFLICT at   1.16 sec ----> 5028 [binary,5027.1,15.1] $ANS(U35).

Length of proof is 13. Level of proof is 5.

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

15 [] -P(i(i(i(i(i(a1,a2),i(a3,a1)),i(i(a1,a2),i(a3,a2))),i(i(a4,a4),i(a5,i(a6,a7)))),i(i(a8,a5),
  i(a6,i(a8,a7)))))|$ANS(U35).
27 [] -P(i(x,y))| -P(x)|P(y).
28 [] P(i(i(x,y),i(i(y,z),i(x,z)))).
30 [] P(i(i(x,i(y,z)),i(y,i(x,z)))).
32 [] P(i(x,x)).
34 [] P(i(i(x,i(x,y)),i(x,y))).
41 [hyper,27,30,30] P(i(x,i(i(y,i(x,z)),i(y,z)))).
44 [hyper,27,30,28] P(i(i(x,y),i(i(z,x),i(z,y)))).
50 [hyper,27,41,32] P(i(i(x,i(i(y,y),z)),i(x,z))).
83 [hyper,27,28,44] P(i(i(i(i(x,y),i(x,z)),u),i(i(y,z),u))).
88 [hyper,27,44,34] P(i(i(x,i(y,i(y,z))),i(x,i(y,z)))).
89 [hyper,27,44,30] P(i(i(x,i(y,i(z,u))),i(x,i(z,i(y,u))))).
90 [hyper,27,44,28] P(i(i(x,i(y,z)),i(x,i(i(z,u),i(y,u))))).
110 [hyper,27,30,50] P(i(x,i(i(x,i(i(y,y),z)),z))).
167 [hyper,27,88,90] P(i(i(i(x,y),i(z,x)),i(i(x,y),i(z,y)))).
254 [hyper,27,110,167] P(i(i(i(i(i(x,y),i(z,x)),i(i(x,y),i(z,y))),i(i(u,u),v)),v)).
4699 [hyper,27,83,89] P(i(i(x,i(y,z)),i(i(u,x),i(y,i(u,z))))).
4806 [hyper,27,44,4699] P(i(i(x,i(y,i(z,u))),i(x,i(i(v,y),i(z,i(v,u)))))).
5027 [hyper,27,4806,254] P(i(i(i(i(i(x,y),i(z,x)),i(i(x,y),i(z,y))),i(i(u,u),i(v,i(w,v6)))),
  i(i(v7,v),i(w,i(v7,v6))))).
```

In summary, in the context of the given 12-step proof, I do marvel at the not-so-obvious places where new proofs hide; indeed, to me this bit of research was startling.

At this point in the notebook, as I implicitly promised when I gave a word of warning, the time has come for a discussion of my attempt to prove from the Ulrich 35-symbol single axiom for *RI* the join of *B*', *C*, *I*, and *W*, in other words to show how I attempted to derive from the Ulrich formula the 4-basis so prominent here. In the spirit of completeness, this bit of research is required to address the perhaps obvious missing item in the study of the subformula strategy in the context of the Ulrich axiom. I was no doubt influenced by various experiments covered here and, possibly, by those presented in another notebook, The Subformula Strategy: Coping with Complex Expressions.

Almost at the same time, I sought the desired proof with a level-saturation approach and with a complexity-preference approach, the latter relying in part on McCune's ratio strategy. Both approaches had some difficulty in proving *W*. However, as the following shows, the complexity-preference approach did succeed, not only in deriving *W*, but in presenting me with a proof of the join of the four.

A Proof, Relying Almost Solely on the Subformula Strategy, of the 4-Basis from the Ulrich Single Axiom

```
----- Otter 3.3g-work, Jan 2005 -----
The process was started by wos on elephant.mcs.anl.gov,
Mon May  5 11:41:06 2008
The command was "otter".  The process ID is 7989.

-----> EMPTY CLAUSE at 2737.16 sec ----> 116127 [hyper,2,268,224,65328,115108] $ANS(bpciw).

Length of proof is 29.  Level of proof is 16.

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

1 [] -P(i(x,y))| -P(x)|P(y).
2 [] -P(i(a,a))| -P(i(i(a,i(b,c)),i(b,i(a,c))))| -P(i(i(a,b),i(i(b,c),i(a,c))))| -P(i(i(a,i(a,b)),i(a,b)))|$ANS(bpciw).
3 [] P(i(i(i(i(u,v),i(w,u)),i(i(u,v),i(w,v))),i(i(x,x),i(y,i(z,v6)))),i(i(v7,y),i(z,i(v7,v6))))).
11 [hyper,1,3,3] P(i(i(x,y),i(i(y,y),i(x,y)))).
12 [hyper,1,11,11] P(i(i(i(i(x,x),i(y,x)),i(i(x,x),i(y,x))),i(i(y,x),i(i(x,x),i(y,x))))).
13 [hyper,1,3,11] P(i(i(x,i(i(y,z),i(u,y))),i(i(y,z),i(x,i(u,z))))).
14 [hyper,1,11,12] P(i(i(i(i(x,y),i(i(y,y),i(x,y))),i(i(x,y),i(i(y,y),i(x,y)))),i(i(i(i(y,y),i(x,y)),
  i(i(y,y),i(x,y))),i(i(x,y),i(i(y,y),i(x,y)))))).
20 [hyper,1,3,13] P(i(i(x,i(i(y,y),i(z,y))),i(z,i(x,y)))).
22 [hyper,1,13,14] P(i(i(i(i(x,x),i(y,x)),i(i(x,x),i(y,x))),i(i(i(i(y,x),i(i(x,x),i(y,x))),i(i(y,x),
  i(i(x,x),i(y,x)))),i(i(y,x),i(i(x,x),i(y,x)))))).
31 [hyper,1,3,20] P(i(i(x,i(i(i(y,z),i(y,z)),i(i(u,u),i(y,z)))),i(y,i(x,z)))).
34 [hyper,1,20,11] P(i(x,i(i(x,y),y))).
45 [hyper,1,31,11] P(i(x,i(i(i(y,y),i(x,z)),z))).
48 [hyper,1,34,34] P(i(i(i(x,i(i(x,y),y)),z),z)).
49 [hyper,1,13,34] P(i(i(x,i(y,x)),i(x,i(y,i(y,x))))).
79 [hyper,1,49,34] P(i(x,i(i(x,x),i(i(x,x),x)))).
155 [hyper,1,20,79] P(i(i(x,x),i(x,x))).
171 [hyper,1,34,155] P(i(i(i(i(x,x),i(x,x)),y),y)).
176 [hyper,1,3,22] P(i(i(x,i(y,z)),i(i(z,z),i(x,i(y,z))))).
224 [hyper,1,3,171] P(i(i(x,i(y,z)),i(y,i(x,z)))).
233 [hyper,1,176,224] P(i(i(i(x,y),i(x,y)),i(i(x,i(z,y)),i(z,i(x,y))))).
252 [hyper,1,224,45] P(i(i(i(x,x),i(y,z)),i(y,z))).
253 [hyper,1,224,34] P(i(i(x,y),i(x,y))).
```

```
266 [hyper,1,31,253] P(i(x,i(i(i(i(x,y),i(x,y)),i(i(z,z),i(x,y))),y))).
268 [hyper,1,252,252] P(i(x,x)).
453 [hyper,1,224,233] P(i(i(x,i(y,z)),i(i(i(x,z),i(x,z)),i(y,i(x,z))))).
587 [hyper,1,266,224] P(i(i(i(i(i(i(x,i(y,z)),i(y,i(x,z))),u),i(i(i(x,i(y,z)),i(y,i(x,z))),u)),i(i(v,v),
  i(i(i(x,i(y,z)),i(y,i(x,z))),u))),u)).
1077 [hyper,1,3,453] P(i(i(x,i(y,z)),i(i(i(y,z),i(u,y)),i(x,i(u,z))))).
1688 [hyper,1,48,1077] P(i(i(i(i(x,y),y),i(z,i(x,y))),i(x,i(z,y)))).
65328 [hyper,1,3,587] P(i(i(x,y),i(i(y,z),i(x,z)))).
65440 [hyper,1,224,65328] P(i(i(x,y),i(i(z,x),i(z,y)))).
66651 [hyper,1,1688,65440] P(i(x,i(i(x,i(x,y)),y))).
115108 [hyper,1,224,66651] P(i(i(x,i(x,y)),i(x,y))).
```

An examination of the input file (to be given almost immediately) that enabled the discovery of this proof shows that, in addition to subformulas, templates were included for the four targets.

An Input File, Relying Almost Solely on the Subformula Strategy, for Deriving the 4-Basis from the Ulrich Single Axiom

```
set(hyper_res).
assign(max_mem,880000).
% assign(max_seconds,2).
% set(sos_queue).
assign(max_weight,32).
assign(change_limit_after,100).
assign(new_max_weight,18).
assign(max_proofs,-1).
assign(pick_given_ratio,4).
assign(max_distinct_vars,5).
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).

weight_list(pick_and_purge).
% Following 8 from Beeson are subformulas of the Ulrich formula.
weight(i(v7,v6),1).
weight(i(i(v7,y),i(z,i(v7,v6))),1).
weight(i(y,i(z,v6)),1).
weight(i(i(x,x),i(y,i(z,v6))),1).
weight(i(i(u,v),i(w,v)),1).
weight(i(i(i(u,v),i(w,u)),i(i(u,v),i(w,v))),1).
weight(i(i(i(i(u,v),i(w,u)),i(i(u,v),i(w,v))),i(i(x,x),i(y,i(z,v6)))),1).
weight(i(i(i(i(i(u,v),i(w,u)),i(i(u,v),i(w,v))),i(i(x,x),i(y,i(z,v6)))),i(i(v7,y),i(z,i(v7,v6)))),1).
% Following are the targets, B' C I W.
weight(P(i(i(u,v),i(i(v,w),i(u,w)))),2). % B'
weight(P(i(i(u,i(v,w)),i(v,i(u,w)))),2). % C
weight(P(i(u,u)),2). % I
weight(P(i(i(u,i(u,v)),i(u,v))),2). % W
end_of_list.
```

```
list(usable).
-P(i(x,y)) | -P(x) | P(y).
-P(i(a,a)) | -P(i(i(a,i(b,c)),i(b,i(a,c)))) | -P(i(i(a,b),i(i(b,c),i(a,c)))) | -P(i(i(a,i(a,b)),i(a,b))) | $ANS(bpciw).
end_of_list.

list(sos).
P(i(i(i(i(i(u,v),i(w,u)),i(i(u,v),i(w,v))),i(i(x,x),i(y,i(z,v6)))),i(i(v7,y),i(z,i(v7,v6))))).
% Ulrich single axiom
end_of_list.

list(passive).
-P(i(a,a)) | $ANS(I).
-P(i(i(a,i(b,c)),i(b,i(a,c)))) | $ANS(C).
-P(i(i(a,b),i(i(b,c),i(a,c)))) | $ANS(B').
-P(i(i(a,i(a,b)),i(a,b))) | $ANS(W).
end_of_list.

list(demodulators).
%  (i(i(x,x),y) = junk).
%  (i(y,i(x,x)) = junk).
(i(x,junk) = junk).
(i(junk,x) = junk).
(P(junk) = $T).
end_of_list.
```

When you are seeking a proof of a conjunction of two or more items, I again stress the wisdom of including, for each member, weight templates (resonators) in the input. By doing so, you sharply increase the likelihood that the program will return a proof of the join, a proof in which no formula or equation occurs more than once.

6. Notes and Observations

At this point, a few notes and observations suffice to close this notebook. One of the more salient comments concerns the seeking of a first proof, for example, of some conjecture not yet shown to be a theorem. The likelihood of success, it seems clear to me, is sharply increased if you have access to and use of an automated reasoning program offering power. If that program does not offer a variety of strategies, the program (from my experiments) is far less useful. The strategies and approaches presented in the various notebooks found on my website may indeed be most helpful when seeking a first proof. This observation may surprise some researchers in that, quite often, my goal is to seek a shorter proof than that in hand or than offered by the literature.

I have commented throughout my writings that, when seeking a shorter proof, the discovery of a shorter so-called subproof does not necessarily signify progress. Two examples nicely illustrate what can go wrong. First, in the context of seeking a proof of the join of two or more elements, a shorter proof of the join may be obtained by accepting proofs, longer than that which can be found, of one or more members; in other words, longer subproofs may be preferred over shorter, in this context. Briefly, the explanation rests with the situation in which the shortest subproofs of the members diverge, share little in common. In contrast, when the subproofs share many steps, than the resulting total proof has a better chance of being shorter than when little sharing is present. For a second example, similar in spirit to that just discussed, when the goal is that of proving a single item (formula or equation), the focus on a shorter subproof of an intermediate step can sometimes lead to the completion of the desired proof, but a proof longer than need be. Indeed, ideally, the steps of a subproof can sometimes be made to do double, triple, or more duty. More on this topic can be found in other notebooks on my website.

The seeking of a first proof was the wellspring for the subformula strategy. Specifically, the goal was to prove from the 4-basis consisting of *B*', *C*, *I*, and *W* d. Ulrich's new single axiom for *RI*, the following.

P(i(i(i(i(i(u,v),i(w,u)),i(i(u,v),i(w,v))),i(i(x,x),i(y,i(z,v6)))),i(i(v7,y),i(z,i(v7,v6))))). % Ulrich single axiom

This impressive formula is the shortest single axiom for *RI* known at this time, May 9 2008. The most obvious obstacle in seeking the desired proof rests with the fact that many reasoning programs—certainly OTTER—prefer to focus on so-called short formulas or equations. How, therefore, would such a program ever focus on formulas sufficiently complex (long) to enable it to complete a proof deriving the given Ulrich axiom? You could, of course, instruct the program, with the use of resonators or Veroff's hints, that items that are complex if measured in symbol count are to be treated as being short. But the problem that interested me asked what you could do if you knew of no resonators or hints to include in the input. In fact, what would you do if the only piece of information you had was the target formula or equation?

No doubt influenced by numerous discussions with R. Overbeek over the years, an answer that occurred to me consisted of the inclusion of some or all of the subformulas of the target. I prefer to rely on the subset where each member contains at least one occurrence of an important function, such as *i* for implication. Clearly, a program can be written to present all subformulas, including the entire target, of a given formula. Both Overbeek and Beeson provided me with what I needed. Success was the result, namely, a proof deriving the Ulrich axiom from the 4-basis featured here. I did in fact misstep, as you see in Section 3, relying on subformulas not all of which are appropriate.

You can of course supplement the use of subformulas with the use of resonators and hints. As I discussed, this study of the subformula strategy did, to my great surprise, in fact eventually lead to finding a proof shorter than weeks of effort had yielded. Research sometimes has charming side-effects: The emphasis on one aspect can yield progress in the context of another. The experiments with the subformula strategy led to the consideration (as discussed) of the Rezus-style 93-symbol formula for *RI*. Eventually, a first proof was found, from what I know, the first such proof relying solely on condensed detachment. Other tests of the value of this new strategy focus on the Lukasiewicz 23-letter single axiom for classical propositional calculus and on the Meredith 21-letter single axiom for that area of logic. They are featured in the notebook titled "The Subformula Strategy: Coping with Complex Expressions".

In addition to the cited items that you can include to guide a program, a wise choice concerns the presence of resonators or hints that correspond to the members of a conjunction when the target is such. Certainly, that is the case for OTTER. The absence of such target correspondents can cause the program to delay substantially in completing a proof of the entire conjunction, even when each of its members has been proved. As a second cousin to such inclusions, I typically include a resonator or hint that corresponds to the target when the target consists of a single formula or equation, assigning to it a smaller value to cause OTTER to consider it and similar items to be treated as having little complexity. The inclusion of weight templates, each assigned a small value (which is in the spirit of the subformula strategy), causes a reasoning program to treat (sometimes) very complex expressions as simple. This treatment can prove vital; after all, when seeking to derive a complex expression, you never know whether even longer (more complex) expressions will be required.

My continued access to Overbeek is indeed a gift of inestimable value. It was he who insisted on the value of seeking proofs, first or shorter, by starting with max_weight assigned the value 1, then 2, and so on. He was the person who found a fine proof for the Meredith single axiom, with an assignment of the value 15 to max_weight. I then, borrowing from his approach, found a proof for the Lukasiewicz single axiom (that of 23 letters) with an assignment of the value 18; 15 did not work. Overbeek points out that the use of the ratio strategy complicates a thorough analysis of what is happening.

The material presented in this notebook extends and improves on some of that which is offered in the notebook The Subformula Strategy. Overbeek offers the following idea, namely, in the spirit of iteration, all nontrivial subformulas of proof steps be adjoined to the weight list when and if items of interest are proved. I offer the following possibility for research. In the context of the aphorism that shorter subproofs do not necessarily a shorter total proof make, you might take proofs of members of a conjunction, or of an intermediate step, that are longer than in hand and attempt to extend such to a new and shorter total proof.

I do invite researchers and students to send proofs to me with the object of finding shorter proofs. I also invite communication by e-mail on one or more topics covered in the notebooks found on this website. Until the next notebook has been completed and its pdf placed in my website, I bid you adieu.