bci.revisited
created 02-20-2009
revised last 06-22-2009

An Astonishing Discovery from Proof Shortening: The Fecundity of the BCI Logic*

Larry Wos

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

1. A Preamble to a Most Unusual Success

Research: how wondrous; how mysterious; and, sometimes, how rewarding! This notebook tells a story of discovery, of improbable occurrence, of research spawning research, a story that requires no expertise to enjoy. The beginning rests with a short phone call from my colleague John Halleck, a call in which he informed me of a most charming proof, obtained by him with his program shotgun. The proof has length 70 (deduced steps, obtained by applying condensed detachment, an inference rule to be defined and illustrated soon), and, amazingly, twenty-eight of the deduced steps are *shortest single axioms* for the *BCI* logic, two due to prior research of C. A. Meredith and twenty-six due to prior research of D. Ulrich. The proof uses as hypotheses three formulas known, respectively, as *B*, *C*, and *I*, each given shortly. The object of the Halleck proof was to deduce the conjunction, or join, of twenty-eight formulas (each of which is a single axiom, to be defined). Halleck had separate proofs of each of the twenty-eight and, as noted, a proof of their conjunction. The idea of having a single proof of the conjunction is that of having in hand a proof of, in this case, all known-at-the-time twenty-eight single axioms that contains no duplicate steps.

A single axiom is a formula so powerful that you need no other formulas to begin a study of the logic in focus. Yes, compared with many areas of logic, the *BCI* logic is indeed fecund in its offering of so many single axioms. Knowledge of that 70-step proof, predictably (and simultaneously at the request of Halleck), caused me to attempt to find a shorter proof, possibly significantly shorter, a proof that uses as hypotheses *B*, *C*, and *I* and that deduces (the conjunction of) all twenty-eight single axioms that Halleck proved with his 70-step proof. The proof I found, with indispensable aid from William McCune's automated reasoning program OTTER, has a most unexpected property, a property that is central to the story to be told. The cited (as-yet undisclosed) property caused Halleck to pursue a line of research that, quite soon, led to the logician D. Ulrich's resuming his research on *BCI* and unearthing a vast amount of treasure. The full story now unfolds.

2. Chapter 1 Optimism

In the following story, you may decide to gloss over various technical details. If this is your decision, you will still find the essence quite surprising, perhaps astounding. The area of concern here is called the *BCI* logic, axiomatized (for this notebook) with the following three formulas, where you can think of "P" as meaning "is provable" and *i* as denoting implies, and you can view "x" and "y" and the like as variables that run over possible formulas.

P(i(i(x,y),i(i(z,x),i(z,y)))). % B, in infix (x->y) -> ((z->x)->(z->y))

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

P(i(i(x,i(y,z)),i(y,i(x,z)))). % C, in infix (x->(y->z)) -> (y->(x->z))
P(i(x,x)). % I, in infix x -> x

The *BCI* logic, as many other areas of logic do, offers single axioms, single formulas from which all of the theorems can be deduced, but no formula can be deduced that is itself not a theorem. (Throughout this notebook, I focus on single axioms, but, to be precise, I focus on shortest single axioms; Ulrich proved that no formula exists that is a single axiom and that relies on strictly fewer than 19 symbols, not counting the predicate symbol, parentheses, or commas.) A typical method for proving that a formula is a single axiom is to first show that it is a theorem and second show that a known basis can be deduced from it. C. A. Meredith offered the following two.

P(i(i(x,i(y,z)),i(i(i(u,u),i(v,y)),i(v,i(x,z))))).
% BCI-1 (x->(y->z)) -> (((u->u)->(v->y))->(v->(x->z)))
P(i(i(i(x,x),i(y,z)),i(i(u,i(z,v)),i(y,i(u,v))))).
% BCI-2 ((x->x)->(y->z)) -> ((u->(z->v))->(y->(u->v)))

Indeed, with an appropriate inference rule for drawing conclusions, you could start with either of the two cited formulas and deduce any theorem of the logic. The rule I used in my study, which is commonly used, is called *condensed detachment*, captured with the use of hyperresolution and the following clause, where "-" denotes logical **not** and " | " denotes logical **or**.

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

And here are the promised definition and illustration. Formally, condensed detachment (in the context of implication) considers two formulas assumed to share no variables in common, *i*(*A*,*B*) (the major premiss) and *C* (the minor premiss), and, if *C* unifies with *A*, yields the formula *D*, where *D* is obtained by applying to *B* a most general unifier of *C* and *A* that does not affect variables in *B* that do not occur in *A*. In other words, to apply the rule successfully, *A* and *C* must unify—a most general substitution must exist that, when applied, causes *A* and *C* to become identical. (For those with some knowledge of various areas of logic, condensed detachment combines an appropriately general rule of substitution with what is called *detachment* or *modus ponens*.) For example, if condensed detachment is applied to

i(x,i(x,i(y,y))) and
i(z,z)

with the second formula playing the role of *C* (the minor premiss), the following is obtained.

i(i(z,z),i(y,y)).

If the roles of the two formulas are reversed and condensed detachment is applied, a copy of the first formula is obtained.

You have seen two single axioms for *BCI*, each due to Meredith, and each present in the Halleck 70-step proof. But you were told that a key 70-step proof contains twenty-eight such formulas. Therefore, you might wonder what the other twenty-six formulas look like and from where they come—and, by the way, when I entered the cast, I was not in possession of the twenty-eight single axioms that I now give. The following are the twenty-six in question (present in the Halleck proof), given to the world by Ulrich on his website at http://web.ics.purdue.edu/~dulrich/BCI-page.htm and shown here in OTTER notation.

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

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

So that you may accompany me on my journey, I shall adhere strictly, more or less, to the story being told here. To begin with, although Halleck sent me his 70-step proof, I paid no attention to the history of each of the seventy deduced steps and, instead, concentrated on the formulas themselves. Indeed, my approach to proof shortening—which was the request made by Halleck, to find a shorter proof than length 70 applications of condensed detachment—does not take into account the parentage of a deduced item. It does, however, very often key on the actual deduced steps of the proof in hand, in this case, the following seventy formulas.

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

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

An inspection of the given seventy formulas reveals a surprising property, namely, that exactly twenty-eight of the formulas rely on five distinct variables. Now, since the proof deduces twenty-eight single axioms and since each of those axioms relies on five distinct variables, the 70-step proof contains no 5-variable formula that is itself not a single axiom. For the record, I made this observation only at the writing of this notebook, a bit after completing the research reported here.

I was almost certain I could find a proof shorter than Halleck's. I chose the goal of finding a 65-step proof, a goal that appeared realistic. My fantasy goal was to find a 56-step proof, a proof half of whose steps were deemed important to logic in that each is a single axiom for *BCI*. I immediately turned, with optimism, to the use of McCune's *ancestor subsumption*, a procedure that automatically attempts to complete shorter proofs. The essence of ancestor subsumption rests with its comparison of two proofs to the same conclusion and, when such are in hand, choosing the strictly shorter of the two to follow. Because I assert that strategy is crucial, I relied on a restriction strategy by limiting, with max_weight, the complexity of retained information. I used a direction strategy, namely, *resonance,* which in the case under discussion relied on the seventy formulas from the Halleck proof. Each of the seventy is preceded by "weight" and followed by an assigned value, informing OTTER that similar items, where all variables are treated as indistinguishable, are to be given *weight* equal to the assigned value. When, for example, an item is assigned the value 2, regardless of how complex it is, similar items, as well as it, are treated as consisting of but two symbols. I used the following input file, a file that can be ignored by the individual who wishes to focus on the essence of the story. (When a percent sign occurs on a line, OTTER treats the rest of the line from that point on as a comment.)

Input File 1

```
set(hyper_res).
assign(max_weight,48).
% assign(max_seconds,600).
% assign(change_limit_after,800).
% assign(new_max_weight,22).
assign(max_proofs,-1).
clear(print_kept).
% set(process_input).
set(ancestor_subsume).
set(back_sub).
% clear(for_sub).
clear(print_back_sub).
clear(print_kept).
clear(print_new_demod).
clear(print_back_demod).
clear(print_back_sub).
% clear(print_given).
assign(max_distinct_vars,8).
assign(pick_given_ratio,2).
assign(max_mem,750000).
assign(report,5400).
set(order_history).
set(input_sos_first).
% set(sos_queue).

weight_list(pick_and_purge).
% Following 70 form the Halleck 70-step proof, the smallest he has, deducing all of the
% 28 single axioms for BCI.
weight(P(i(i(x,i(y,i(z,u))),i(x,i(z,i(y,u))))),2).
weight(P(i(i(x,y),i(z,i(i(z,x),y)))),2).
weight(P(i(x,i(i(x,i(y,z)),i(u,i(i(u,y),z))))),2).
weight(P(i(i(i(x,x),i(y,z)),i(u,i(i(u,y),z)))),2).
weight(P(i(i(x,y),i(i(y,z),i(x,z)))),2).
weight(P(i(i(x,i(y,z)),i(x,i(i(z,u),i(y,u))))),2).
weight(P(i(i(i(x,x),i(y,z)),i(i(i(i(u,y),z),v),i(u,v)))),2).
weight(P(i(x,i(i(x,y),y))),2).
weight(P(i(i(i(x,x),y),y)),2).
weight(P(i(x,i(i(i(y,y),i(x,z)),z))),2).
weight(P(i(i(i(i(i(x,x),i(y,z)),z),u),i(y,u))),2).
weight(P(i(i(i(i(i(x,x),i(y,z)),z),u),i(i(u,v),i(y,v)))),2).
weight(P(i(i(x,i(y,z)),i(i(i(x,z),u),i(y,u)))),2).
weight(P(i(i(x,y),i(i(i(z,z),x),y))),2).
```

weight(P(i(i(i(i(i(x,x),y),z),u),i(i(y,z),u))),2).
weight(P(i(i(x,i(y,z)),i(i(i(i(u,u),x),z),v),i(y,v)))),2).
weight(P(i(i(i(i(x,y),i(x,z)),u),i(i(y,z),u))),2).
weight(P(i(i(x,y),i(i(i(z,z),i(u,x)),i(u,y)))),2).
weight(P(i(i(i(x,i(y,z)),u),i(i(y,i(x,z)),u))),2).
weight(P(i(i(x,i(y,z)),i(i(i(u,u),i(v,y)),i(v,i(x,z))))),2).
weight(P(i(x,i(i(y,i(x,z)),i(y,z)))),2).
weight(P(i(i(x,i(i(y,y),z)),i(x,z))),2).
weight(P(i(i(x,i(i(y,y),i(z,u))),i(z,i(x,u)))),2).
weight(P(i(i(i(x,y),z),i(i(u,y),i(i(x,u),z)))),2).
weight(P(i(i(x,i(i(y,y),i(z,u))),i(i(v,x),i(z,i(v,u))))),2).
weight(P(i(i(x,y),i(x,i(i(y,z),z)))),2).
weight(P(i(i(i(x,x),i(y,z)),i(y,i(i(z,u),u)))),2).
weight(P(i(i(i(x,x),i(y,z)),i(i(i(i(z,u),u),v),i(y,v)))),2).
weight(P(i(i(i(i(x,y),i(z,y)),u),i(i(z,x),u))),2).
weight(P(i(i(x,y),i(i(y,i(z,u)),i(z,i(x,u))))),2).
weight(P(i(i(i(x,y),z),i(i(x,i(i(u,u),y)),z))),2).
weight(P(i(i(x,i(i(y,y),z)),i(i(z,i(u,v)),i(u,i(x,v))))),2).
weight(P(i(i(x,i(y,z)),i(x,i(i(u,y),i(u,z))))),2).
weight(P(i(i(x,i(y,z)),i(i(u,y),i(u,i(x,z))))),2).
weight(P(i(i(x,y),i(i(z,i(y,u)),i(x,i(z,u))))),2).
weight(P(i(i(i(x,x),i(y,z)),i(i(u,i(z,v)),i(y,i(u,v))))),2).
weight(P(i(i(x,i(y,z)),i(i(u,x),i(y,i(u,z))))),2).
weight(P(i(i(i(x,x),i(y,i(z,u))),i(i(v,y),i(z,i(v,u))))),2).
weight(P(i(i(i(x,x),i(y,z)),i(i(z,i(u,v)),i(u,i(y,v))))),2).
weight(P(i(i(i(i(x,y),y),z),i(x,z))),2).
weight(P(i(i(i(x,x),i(i(i(y,z),z),u)),i(y,u))),2).
weight(P(i(i(i(x,x),i(i(i(y,z),z),u)),i(i(u,v),i(y,v)))),2).
weight(P(i(i(x,i(i(y,y),z)),i(u,i(i(z,i(u,v)),i(x,v))))),2).
weight(P(i(i(x,y),i(i(i(i(z,u),u),x),i(z,y)))),2).
weight(P(i(i(i(x,x),i(y,z)),i(i(i(i(u,v),v),y),i(u,z)))),2).
weight(P(i(x,i(i(x,i(y,z)),i(i(u,y),i(u,z))))),2).
weight(P(i(i(i(x,x),i(y,z)),i(i(u,y),i(u,z)))),2).
weight(P(i(i(x,y),i(i(i(z,z),i(y,u)),i(x,u)))),2).
weight(P(i(i(i(x,i(i(x,y),z)),u),i(i(y,z),u))),2).
weight(P(i(i(x,y),i(i(i(z,z),i(i(i(u,x),y),v)),i(u,v)))),2).
weight(P(i(i(x,y),i(i(y,i(i(z,z),i(u,v))),i(u,i(x,v))))),2).
weight(P(i(i(i(x,i(i(y,z),z)),u),i(i(x,y),u))),2).
weight(P(i(i(x,y),i(i(i(z,z),i(i(i(y,u),u),v)),i(x,v)))),2).
weight(P(i(i(x,i(y,z)),i(i(i(u,u),i(v,x)),i(y,i(v,z))))),2).
weight(P(i(i(x,y),i(i(i(z,z),i(y,i(u,v))),i(u,i(x,v))))),2).
weight(P(i(i(i(x,y),z),i(i(x,i(u,y)),i(u,z)))),2).
weight(P(i(i(x,i(y,z)),i(y,i(i(i(u,u),i(z,v)),i(x,v))))),2).
weight(P(i(i(i(x,x),i(y,z)),i(u,i(i(z,i(u,v)),i(y,v))))),2).
weight(P(i(i(i(x,y),z),i(i(i(i(x,u),u),y),z))),2).
weight(P(i(i(i(i(x,y),y),z),i(i(i(u,u),i(z,v)),i(x,v)))),2).
weight(P(i(i(x,y),i(i(z,i(u,x)),i(u,i(z,y))))),2).
weight(P(i(i(i(x,x),i(y,z)),i(i(u,i(v,y)),i(v,i(u,z))))),2).
weight(P(i(i(x,i(y,z)),i(i(z,u),i(y,i(x,u))))),2).
weight(P(i(i(x,i(i(y,y),i(z,u))),i(i(u,v),i(z,i(x,v))))),2).
weight(P(i(x,i(i(i(y,y),i(z,u)),i(i(u,i(x,v)),i(z,v))))),2).
weight(P(i(i(x,y),i(z,i(i(i(u,u),i(y,i(z,v))),i(x,v))))),2).
weight(P(i(x,i(i(y,i(x,z)),i(i(i(u,u),i(z,v)),i(y,v))))),2).

```
weight(P(i(i(x,i(y,i(z,u))),i(z,i(x,i(y,u))))),2).
weight(P(i(x,i(i(i(y,y),i(z,i(x,u))),i(i(v,z),i(v,u))))),2).
weight(P(i(i(x,i(y,z)),i(i(i(u,u),i(z,v)),i(y,i(x,v))))),2).
end_of_list.

list(usable).
-P(i(x,y)) | -P(x) | P(y).
-P(i(i(a1,i(a2,a3)),i(i(i(a4,a4),i(a5,a2)),i(a5,i(a1,a3))))) |
-P(i(i(i(a1,a1),i(a2,a3)),i(i(a4,i(a3,a5)),i(a2,i(a4,a5))))) |
-P(i(i(a1,a2),i(i(i(a3,a3),i(i(i(a2,a4),a4),a5)),i(a1,a5)))) |
-P(i(i(a1,a2),i(i(a2,i(i(a3,a3),i(a4,a5))),i(a4,i(a1,a5))))) |
-P(i(i(i(i(i(a1,a1),i(a2,a3)),a3),a4),i(i(a4,a5),i(a2,a5)))) |
-P(i(i(i(i(a1,a2),a2),a3),i(i(i(a4,a4),i(a3,a5)),i(a1,a5)))) |
-P(i(i(a1,i(a2,a3)),i(i(i(i(a4,a4),a1),a3),a5),i(a2,a5)))) |
-P(i(i(a1,i(a2,a3)),i(i(i(a4,a4),i(a3,a5)),i(a2,i(a1,a5))))) |
-P(i(i(i(a1,a1),i(i(i(a2,a3),a3),a4)),i(i(a4,a5),i(a2,a5)))) |
-P(i(i(i(a1,a1),i(a2,a3)),i(a4,i(i(a3,i(a4,a5)),i(a2,a5))))) |
-P(i(i(i(a1,a1),i(a2,a3)),i(i(i(i(a3,a4),a4),a5),i(a2,a5)))) |
-P(i(i(i(a1,a1),i(a2,a3)),i(i(a3,i(a4,a5)),i(a4,i(a2,a5))))) |
-P(i(i(a1,i(i(a2,a2),i(a3,a4))),i(i(a4,a5),i(a3,i(a1,a5))))) |
-P(i(i(a1,i(i(a2,a2),a3)),i(a4,i(i(a3,i(a4,a5)),i(a1,a5))))) |
-P(i(i(a1,a2),i(i(i(a3,a3),i(a2,i(a4,a5))),i(a4,i(a1,a5))))) |
-P(i(i(a1,i(a2,a3)),i(a2,i(i(i(a4,a4),i(a3,a5)),i(a1,a5))))) |
-P(i(a1,i(i(i(a2,a2),i(a3,a4)),i(i(a4,i(a1,a5)),i(a3,a5))))) |
-P(i(a1,i(i(a2,i(a1,a3)),i(i(i(a4,a4),i(a3,a5)),i(a2,a5))))) |
-P(i(i(a1,a2),i(a3,i(i(i(a4,a4),i(a2,i(a3,a5))),i(a1,a5))))) |
-P(i(i(i(a1,a1),i(a2,a3)),i(i(i(i(a4,a5),a5),a2),i(a4,a3)))) |
-P(i(a1,i(i(i(a2,a2),i(a3,i(a1,a4))),i(i(a5,a3),i(a5,a4))))) |
-P(i(i(i(a1,a1),i(a2,i(a3,a4))),i(i(a5,a2),i(a3,i(a5,a4))))) |
-P(i(i(a1,a2),i(i(i(a3,a3),i(i(i(a4,a1),a2),a5)),i(a4,a5)))) |
-P(i(i(a1,i(i(a2,a2),i(a3,a4))),i(i(a5,a1),i(a3,i(a5,a4))))) |
-P(i(i(i(a1,a1),i(a2,a3)),i(i(i(i(a4,a2),a3),a5),i(a4,a5)))) |
-P(i(i(i(a1,a1),i(a2,a3)),i(i(a4,i(a5,a2)),i(a5,i(a4,a3))))) |
-P(i(i(a1,i(i(a2,a2),a3)),i(i(a3,i(a4,a5)),i(a4,i(a1,a5))))) |
-P(i(i(a1,i(a2,a3)),i(i(i(a4,a4),i(a5,a1)),i(a2,i(a5,a3))))) | $ANS(TARGALL).
%  -P(i(i(a1,i(b,a2)),i(b,i(a1,a2)))) | -P(i(a1,a1)) | -P(i(i(a1,b),i(i(a2,a1),i(a2,b)))) | $ANS(all).  % BCI
end_of_list.

list(sos).
P(i(i(x,y),i(i(z,x),i(z,y)))).  % B
P(i(i(x,i(y,z)),i(y,i(x,z)))).  % C
P(i(x,x)).  % I
end_of_list.

list(passive).
%  %  Following 28 are single axioms for BCI.
%  -P(i(i(a1,i(a2,a3)),i(i(i(a4,a4),i(a5,a2)),i(a5,i(a1,a3))))) | $ANS(TARG).
%  -P(i(i(i(a1,a1),i(a2,a3)),i(i(a4,i(a3,a5)),i(a2,i(a4,a5))))) | $ANS(TARG).
%  -P(i(i(a1,a2),i(i(i(a3,a3),i(i(i(a2,a4),a4),a5)),i(a1,a5)))) | $ANS(TARG).
%  -P(i(i(a1,a2),i(i(a2,i(i(a3,a3),i(a4,a5))),i(a4,i(a1,a5))))) | $ANS(TARG).
%  -P(i(i(i(i(i(a1,a1),i(a2,a3)),a3),a4),i(i(a4,a5),i(a2,a5)))) | $ANS(TARG).
%  -P(i(i(i(i(a1,a2),a2),a3),i(i(i(a4,a4),i(a3,a5)),i(a1,a5)))) | $ANS(TARG).
%  -P(i(i(a1,i(a2,a3)),i(i(i(i(a4,a4),a1),a3),a5),i(a2,a5)))) | $ANS(TARG).
```

```
% -P(i(i(a1,i(a2,a3)),i(i(i(a4,a4),i(a3,a5)),i(a2,i(a1,a5))))) | $ANS(TARG).
% -P(i(i(i(a1,a1),i(i(i(a2,a3),a3),a4)),i(i(a4,a5),i(a2,a5)))) | $ANS(TARG).
% -P(i(i(i(a1,a1),i(a2,a3)),i(a4,i(i(a3,i(a4,a5)),i(a2,a5))))) | $ANS(TARG).
% -P(i(i(i(a1,a1),i(a2,a3)),i(i(i(i(a3,a4),a4),a5),i(a2,a5)))) | $ANS(TARG).
% -P(i(i(i(a1,a1),i(a2,a3)),i(i(a3,i(a4,a5)),i(a4,i(a2,a5))))) | $ANS(TARG).
% -P(i(i(a1,i(i(a2,a2),i(a3,a4))),i(i(a4,a5),i(a3,i(a1,a5))))) | $ANS(TARG).
% -P(i(i(a1,i(i(a2,a2),a3)),i(a4,i(i(a3,i(a4,a5)),i(a1,a5))))) | $ANS(TARG).
% -P(i(i(a1,a2),i(i(i(a3,a3),i(a2,i(a4,a5))),i(a4,i(a1,a5))))) | $ANS(TARG).
% -P(i(i(a1,i(a2,a3)),i(a2,i(i(i(a4,a4),i(a3,a5)),i(a1,a5))))) | $ANS(TARG).
% -P(i(a1,i(i(i(a2,a2),i(a3,a4)),i(i(a4,i(a1,a5)),i(a3,a5))))) | $ANS(TARG).
% -P(i(a1,i(i(a2,i(a1,a3)),i(i(i(a4,a4),i(a3,a5)),i(a2,a5))))) | $ANS(TARG).
% -P(i(i(a1,a2),i(a3,i(i(i(a4,a4),i(a2,i(a3,a5))),i(a1,a5))))) | $ANS(TARG).
% -P(i(i(i(a1,a1),i(a2,a3)),i(i(i(i(a4,a5),a5),a2),i(a4,a3)))) | $ANS(TARG).
% -P(i(a1,i(i(i(a2,a2),i(a3,i(a1,a4))),i(i(a5,a3),i(a5,a4))))) | $ANS(TARG).
% -P(i(i(i(a1,a1),i(a2,i(a3,a4))),i(i(a5,a2),i(a3,i(a5,a4))))) | $ANS(TARG).
% -P(i(i(a1,a2),i(i(i(a3,a3),i(i(i(a4,a1),a2),a5)),i(a4,a5)))) | $ANS(TARG).
% -P(i(i(a1,i(i(a2,a2),i(a3,a4))),i(i(a5,a1),i(a3,i(a5,a4))))) | $ANS(TARG).
% -P(i(i(i(a1,a1),i(a2,a3)),i(i(i(i(a4,a2),a3),a5),i(a4,a5)))) | $ANS(TARG).
% -P(i(i(i(a1,a1),i(a2,a3)),i(i(a4,i(a5,a2)),i(a5,i(a4,a3))))) | $ANS(TARG).
% -P(i(i(a1,i(i(a2,a2),a3)),i(i(a3,i(a4,a5)),i(a4,i(a1,a5))))) | $ANS(TARG).
% -P(i(i(a1,i(a2,a3)),i(i(i(a4,a4),i(a5,a1)),i(a2,i(a5,a3))))) | $ANS(TARG).
end_of_list.

% Following purges unwanted formulas.
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.
```

The experiment yielded five proofs—yes, one of the fine features of OTTER is its ability, when instructed to do so, to seek many proofs in a single experiment. The first proof was completed in approximately 56 CPU-seconds and the fifth in approximately 6100 CPU-seconds. However, in that—at least when my search for a shorter proof commences—ancestor subsumption almost always yields progress, you can imagine my surprise to learn that the five proofs were of respective lengths 75, 76, 75, 79, and 79. The failure to make progress—not even returning Halleck's 70-step proof—does not rest with McCune's powerful and useful procedure of ancestor subsumption. Instead, the problem of finding a shorter proof was harder than it appeared to me. So what move could I make next to seek the desired shorter proof of an unspecified length?

3. Chapter 2, Progress

Based on thousands of experiments, the next action to take (in search of the goal of a shorter proof) was clear to me. The way it works is the following. Whether the reliance solely on ancestor subsumption succeeds or not, the next phase of my approach brings into play a nonstandard use of *demodulation*, yes, coupled with ancestor subsumption. Demodulation is ordinarily used to simplify and canonicalize items, replacing, for example, $0 + a$ by a. In the given context, however, demodulation is used to purge unwanted items by (in effect) demodulating, or rewriting, such to "junk" and thus attempt to take advantage of an interesting property that is often present in a proof. For example, if you wished to prevent OTTER from retaining the first formula found in the following, you could include this.

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

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

This inclusion would prevent the program from retaining, and hence using, the first formula found in the example. If that formula were a step of, say, a 30-step proof, then the program would be given the assignment of completing a different proof, perhaps one of length strictly less than 30.

The property is concerned with the number of sets of parents a formula can have. In particular, in contrast to a given cub that has a unique mother and father bear, a given formula can be derived (with, say, condensed detachment) from various pairs of parents. For an illustration of what can happen, consider a (hypothetical) 30-step proof in which the twelfth derived step has as cited parents the fourth and sixth derived formulas, the twentieth has as cited parents the fourth and eleventh, and no other derived step has the fourth formula as a cited parent. Further, let this 30-step proof be such that the twelfth derived step can be obtained from the seventh and ninth steps as parents and the twentieth can be obtained from the fourteenth and fifteenth. Now, if you replace the given 30-step proof by a new proof obtained by deleting the fourth of the derived steps of the 30-step proof coupled with modifications to the history of the new eleventh and nineteenth steps (based on the properties just given), you will have a 29-step proof. With the cited nonstandard use of demodulation, you can ask OTTER (or whatever reasoning program you are using, if it has what is needed) to systematically block the use of each of the derived steps of the proof you have in hand. The goal is, of course, to have the program find a proof that is shorter, by one or more steps, than the proof in hand. It often works, as you will see from the experiments described here in Chapter 2 of the story being told.

My approach was to adjoin a demodulator when I found that its use produced a shorter proof than that in hand. Iteration was the key. After six successes, I had found a 61-step proof, essentially using the given input file with the following modifications. I added in the pick_and_purge list weight(junk,1000). to purge any deduced conclusion that contained junk. More important, I included the following demodulator list.

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

(By the way, for the curious, before demodulation was brought into play, a breadth-first search yielded nothing of interest.)

The new 61-step proof of the join of the earlier-cited twenty-eight-step single axiom contained the following six formulas not among the Halleck 70.

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

A glance at the listed six reveals an interesting property—at least when this story concludes, the property is of interest. Specifically, four of the six new formulas each rely on five distinct variables, as each of the twenty-eight single axioms does. Also, sharing a property with the single axioms in focus, the four each have length 19, if the predicate symbol, parentheses, and commas are not counted.

Because I believe it could matter to at least part of the world, I note here that my account might not precisely match history. The essence of what I report in this notebook is, nevertheless, accurate. I apparently took the modified input file and continued to block various formulas one at a time with demodulation. I suspect I focused at this point on blocking steps of the 61-step proof. With the following three additional demodulators, OTTER returned to me a 58-step proof.

```
(P(i(i(i(x,y),z),i(i(x,i(u,y)),i(u,z)))) = junk).
(P(i(i(i(x,x),i(y,z)),i(i(u,y),i(u,z)))) = junk).
(P(i(x,i(i(i(y,y),i(x,z)),z))) = junk).
```

Just as a review, you see that removing various formulas from possible parentage leads to progress, at least in some cases, permitting other formulas, often already in the proof in hand, to take over the responsibility of the removed formulas. The following formulas are in the 58-step proof and not among the Halleck 70.

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

The following four are in the 58-step proof and not in the 61-step proof.

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

The 58-step proof contains two formulas that are both not among the Halleck 70 and yet resemble 5-variable single axioms.

I am omitting from this narrative those experiments that did not evince progress. I now give a second input file that might prove useful for verifying my claims and for research of another nature. In that file, you will see that I turned from the use of the resonance strategy to the use of R. Veroff's *hints strategy*, a powerful strategy indeed. The hints strategy focuses on items subsuming and subsumable by items in the hints list, and not (as with resonance) on the functional shape of an item where all variables are treated as indistinguishable.

Input File 2

```
set(hyper_res).
assign(max_weight,14).
assign(max_seconds,6).
% assign(change_limit_after,800).
% assign(new_max_weight,22).
assign(max_proofs,-1).
clear(print_kept).
% set(process_input).
set(ancestor_subsume).
set(back_sub).
% clear(for_sub).
clear(print_back_sub).
clear(print_kept).
```

```
clear(print_new_demod).
clear(print_back_demod).
clear(print_back_sub).
% clear(print_given).
assign(max_distinct_vars,8).
assign(pick_given_ratio,2).
assign(max_mem,750000).
assign(report,5400).
set(order_history).
set(input_sos_first).
%  set(sos_queue).
% set(sos_stack).
assign(bsub_hint_wt,1).
set(keep_hint_subsumers).
assign(heat,0).
% assign(dynamic_heat_weight,2).

weight_list(pick_and_purge).
weight(P(i(i(x,y),i(z,i(i(z,x),y)))),1000).
weight(junk,1000).
end_of_list.

list(usable).
-P(i(x,y)) | -P(x) | P(y).
-P(i(i(a1,i(a2,a3)),i(i(i(a4,a4),i(a5,a2)),i(a5,i(a1,a3))))) |
-P(i(i(i(a1,a1),i(a2,a3)),i(i(a4,i(a3,a5)),i(a2,i(a4,a5))))) |
-P(i(i(a1,a2),i(i(i(a3,a3),i(i(i(a2,a4),a4),a5)),i(a1,a5)))) |
-P(i(i(a1,a2),i(i(a2,i(i(a3,a3),i(a4,a5))),i(a4,i(a1,a5))))) |
-P(i(i(i(i(i(a1,a1),i(a2,a3)),a3),a4),i(i(a4,a5),i(a2,a5)))) |
-P(i(i(i(i(a1,a2),a2),a3),i(i(i(a4,a4),i(a3,a5)),i(a1,a5)))) |
-P(i(i(a1,i(a2,a3)),i(i(i(i(i(a4,a4),a1),a3),a5),i(a2,a5)))) |
-P(i(i(a1,i(a2,a3)),i(i(i(a4,a4),i(a3,a5)),i(a2,i(a1,a5))))) |
-P(i(i(i(a1,a1),i(i(i(a2,a3),a3),a4)),i(i(a4,a5),i(a2,a5)))) |
-P(i(i(i(a1,a1),i(a2,a3)),i(a4,i(i(a3,i(a4,a5)),i(a2,a5))))) |
-P(i(i(i(a1,a1),i(a2,a3)),i(i(i(i(a3,a4),a4),a5),i(a2,a5)))) |
-P(i(i(i(a1,a1),i(a2,a3)),i(i(a3,i(a4,a5)),i(a4,i(a2,a5))))) |
-P(i(i(a1,i(i(a2,a2),i(a3,a4))),i(i(a4,a5),i(a3,i(a1,a5))))) |
-P(i(i(a1,i(i(a2,a2),a3)),i(a4,i(i(a3,i(a4,a5)),i(a1,a5))))) |
-P(i(i(a1,a2),i(i(i(a3,a3),i(a2,i(a4,a5))),i(a4,i(a1,a5))))) |
-P(i(i(a1,i(a2,a3)),i(a2,i(i(i(a4,a4),i(a3,a5)),i(a1,a5))))) |
-P(i(a1,i(i(i(a2,a2),i(a3,a4)),i(i(a4,i(a1,a5)),i(a3,a5))))) |
-P(i(a1,i(i(a2,i(a1,a3)),i(i(i(a4,a4),i(a3,a5)),i(a2,a5))))) |
-P(i(i(a1,a2),i(a3,i(i(i(a4,a4),i(a2,i(a3,a5))),i(a1,a5))))) |
-P(i(i(i(a1,a1),i(a2,a3)),i(i(i(i(a4,a5),a5),a2),i(a4,a3)))) |
-P(i(a1,i(i(i(a2,a2),i(a3,i(a1,a4))),i(i(a5,a3),i(a5,a4))))) |
-P(i(i(i(a1,a1),i(a2,i(a3,a4))),i(i(a5,a2),i(a3,i(a5,a4))))) |
-P(i(i(a1,a2),i(i(i(a3,a3),i(i(i(a4,a1),a2),a5)),i(a4,a5)))) |
-P(i(i(a1,i(i(a2,a2),i(a3,a4))),i(i(a5,a1),i(a3,i(a5,a4))))) |
-P(i(i(i(a1,a1),i(a2,a3)),i(i(i(i(a4,a2),a3),a5),i(a4,a5)))) |
-P(i(i(i(a1,a1),i(a2,a3)),i(i(a4,i(a5,a2)),i(a5,i(a4,a3))))) |
-P(i(i(a1,i(i(a2,a2),a3)),i(i(a3,i(a4,a5)),i(a4,i(a1,a5))))) |
-P(i(i(a1,i(a2,a3)),i(i(i(a4,a4),i(a5,a1)),i(a2,i(a5,a3))))) | $ANS(TARGALL).
%  -P(i(i(a1,i(b,a2)),i(b,i(a1,a2)))) | -P(i(a1,a1)) | -P(i(i(a1,b),i(i(a2,a1),i(a2,b)))) |
```

```
%  $ANS(all).  % BCI
end_of_list.

list(sos).
P(i(i(x,y),i(i(z,x),i(z,y)))).  % B
P(i(i(x,i(y,z)),i(y,i(x,z)))).  % C
P(i(x,x)).  % I
end_of_list.

list(passive).
end_of_list.

% Following purges unwanted formulas.
list(demodulators).
(P(i(i(x,y),i(i(z,i(y,u)),i(x,i(z,u))))) = junk).
(P(i(i(x,i(y,z)),i(x,i(i(i(u,u),i(z,v)),i(y,v))))) = junk).
(P(i(i(x,y),i(i(y,i(z,u)),i(z,i(x,u))))) = 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.

list(hints).
% Following 58 purport to prove the join of the 28 singles for BCI.
P(i(i(x,i(y,z)),i(x,i(i(u,y),i(u,z))))).
P(i(i(x,i(y,i(z,u))),i(x,i(z,i(y,u))))).
P(i(i(x,y),i(i(y,z),i(x,z)))).
P(i(x,i(i(x,y),y))).
P(i(i(x,i(y,z)),i(i(u,y),i(u,i(x,z))))).
P(i(i(x,i(y,i(z,u))),i(z,i(x,i(y,u))))).
P(i(i(x,i(y,z)),i(i(u,y),i(x,i(u,z))))).
P(i(i(i(i(x,y),i(z,y)),u),i(i(z,x),u))).
P(i(i(x,i(y,z)),i(x,i(i(z,u),i(y,u))))).
P(i(i(i(x,i(y,z)),u),i(i(y,i(x,z)),u))).
P(i(i(i(i(x,y),i(x,z)),u),i(i(y,z),u))).
P(i(i(i(i(x,y),y),z),i(x,z))).
P(i(i(x,y),i(x,i(i(y,z),z)))).
P(i(i(i(x,x),y),y)).
P(i(i(x,y),i(i(z,i(y,u)),i(x,i(z,u))))).
P(i(i(x,y),i(i(y,i(z,u)),i(z,i(x,u))))).
P(i(i(x,i(y,z)),i(i(i(x,z),u),i(y,u)))).
P(i(i(x,i(y,z)),i(i(u,x),i(y,i(u,z))))).
P(i(i(i(x,y),z),i(i(i(i(x,u),u),y),z))).
P(i(i(x,y),i(i(i(i(y,z),z),u),i(x,u)))).
P(i(i(x,y),i(i(i(z,z),i(y,u)),i(x,u)))).
P(i(i(x,y),i(i(i(z,z),x),y))).
P(i(i(x,i(y,z)),i(i(i(u,u),y),i(x,z)))).
P(i(i(x,y),i(i(i(i(z,x),y),u),i(z,u)))).
P(i(i(i(i(i(x,x),i(y,z)),z),u),i(y,u))).
P(i(i(x,y),i(i(z,i(u,x)),i(u,i(z,y))))).
P(i(i(i(i(x,y),y),z),i(i(i(u,u),i(z,v)),i(x,v)))).
```

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

You will note that I changed the focus from an emphasis on the Halleck 70 to a focus on the derived steps of the earlier-cited 58-step proof. Again, demodulation was relied upon to block the use of formulas. The result was the discovery, after iteration as usual, of a 54-step proof. The even newer 54-step proof contains the following five formulas, among its deduced steps, that are not among the Halleck 70.

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

On the other hand, all of the 54 deduced steps are present in the 58-step proof. As for 5-variable formulas, the 54-step proof (as you might deduce) contains but one not among the Halleck 70, indeed, not among the earlier-cited twenty-eight single axioms. That formula will become a star in this story.

This chapter closes with one additional success, commentary on goals, and a conjecture. One final experiment in this series deviated slightly from that yielding the 54-step proof. The max_weight was assigned the value 19, rather than 14; the pick_given_ratio was assigned the value 10, rather than 2; and the following demodulator was included.

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

McCune's *pick_given_ratio strategy* combines choosing where next to focus complexity preference with breadth-first; with the value 10, ten items are chosen based on complexity preference, 1 by breadth-first, 10, 1, and the like. This last experiment yielded a 53-step proof, the following, all of whose steps are among the 54 of the 54-step proof.

A Remarkable 53-Step Proof

----- Otter 3.3g-work, Jan 2005 -----
The process was started by wos on octopus.mcs.anl.gov,
Tue Feb 17 16:19:56 2009
The command was "otter". The process ID is 28452.

-----> EMPTY CLAUSE at 0.10 sec ----> 310 [hyper,2,279,230,219,282,204,298,234,249,218,274,190,
227,262,301,239,293,273,292,286,309,238,191,308,237,198,212,272,275] $ANS(TARGALL).

Length of proof is 53. Level of proof is 8.

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

1 1 [] -P(i(x,y)) | -P(x) | P(y).
2 [] -P(i(i(a1,i(a2,a3)),i(i(i(a4,a4),i(a5,a2)),i(a5,i(a1,a3))))) |
-P(i(i(i(a1,a1),i(a2,a3)),i(i(a4,i(a3,a5)),i(a2,i(a4,a5))))) |
-P(i(i(a1,a2),i(i(i(a3,a3),i(i(i(a2,a4),a4),a5)),i(a1,a5)))) |
-P(i(i(a1,a2),i(i(a2,i(i(a3,a3),i(a4,a5))),i(a4,i(a1,a5))))) |
-P(i(i(i(i(i(a1,a1),i(a2,a3)),a3),a4),i(i(a4,a5),i(a2,a5)))) |
-P(i(i(i(i(a1,a2),a2),a3),i(i(i(a4,a4),i(a3,a5)),i(a1,a5)))) |
-P(i(i(a1,i(a2,a3)),i(i(i(i(a4,a4),a1),a3),a5),i(a2,a5)))) |
-P(i(i(a1,i(a2,a3)),i(i(i(a4,a4),i(a3,a5)),i(a2,i(a1,a5))))) |
-P(i(i(i(a1,a1),i(i(i(a2,a3),a3),a4)),i(i(a4,a5),i(a2,a5)))) |
-P(i(i(i(a1,a1),i(a2,a3)),i(a4,i(i(a3,i(a4,a5)),i(a2,a5))))) |
-P(i(i(i(a1,a1),i(a2,a3)),i(i(i(i(a3,a4),a4),a5),i(a2,a5)))) |
-P(i(i(i(a1,a1),i(a2,a3)),i(i(a3,i(a4,a5)),i(a4,i(a2,a5))))) |
-P(i(i(a1,i(i(a2,a2),i(a3,a4))),i(i(a4,a5),i(a3,i(a1,a5))))) |
-P(i(i(a1,i(i(a2,a2),a3)),i(a4,i(i(a3,i(a4,a5)),i(a1,a5))))) |
-P(i(i(a1,a2),i(i(i(a3,a3),i(a2,i(a4,a5))),i(a4,i(a1,a5))))) |
-P(i(i(a1,i(a2,a3)),i(a2,i(i(i(a4,a4),i(a3,a5)),i(a1,a5))))) |
-P(i(a1,i(i(i(a2,a2),i(a3,a4)),i(i(a4,i(a1,a5)),i(a3,a5))))) |
-P(i(a1,i(i(a2,i(a1,a3)),i(i(i(a4,a4),i(a3,a5)),i(a2,a5))))) |
-P(i(i(a1,a2),i(a3,i(i(i(a4,a4),i(a2,i(a3,a5))),i(a1,a5))))) |
-P(i(i(i(a1,a1),i(a2,a3)),i(i(i(i(a4,a5),a5),a2),i(a4,a3)))) |
-P(i(a1,i(i(i(a2,a2),i(a3,i(a1,a4))),i(i(a5,a3),i(a5,a4))))) |
-P(i(i(i(a1,a1),i(a2,i(a3,a4))),i(i(a5,a2),i(a3,i(a5,a4))))) |
-P(i(i(a1,a2),i(i(i(a3,a3),i(i(i(a4,a1),a2),a5)),i(a4,a5)))) |
-P(i(i(a1,i(i(a2,a2),i(a3,a4))),i(i(a5,a1),i(a3,i(a5,a4))))) |
-P(i(i(i(a1,a1),i(a2,a3)),i(i(i(i(a4,a2),a3),a5),i(a4,a5)))) |
-P(i(i(i(a1,a1),i(a2,a3)),i(i(a4,i(a5,a2)),i(a5,i(a4,a3))))) |
-P(i(i(a1,i(i(a2,a2),a3)),i(i(a3,i(a4,a5)),i(a4,i(a1,a5))))) |
-P(i(i(a1,i(a2,a3)),i(i(i(a4,a4),i(a5,a1)),i(a2,i(a5,a3))))) | $ANS(TARGALL).
3 [] P(i(i(x,y),i(i(z,x),i(z,y)))).
4 [] P(i(i(x,i(y,z)),i(y,i(x,z)))).
5 [] P(i(x,x)).
71 [hyper,1,3,3] P(i(i(x,i(y,z)),i(x,i(i(u,y),i(u,z))))).
73 [hyper,1,3,4] P(i(i(x,i(y,i(z,u))),i(x,i(z,i(y,u))))).
74 [hyper,1,4,3] P(i(i(x,y),i(i(y,z),i(x,z)))).
75 [hyper,1,4,5] P(i(x,i(i(x,y),y))).

77 [hyper,1,71,4] P(i(i(x,i(y,z)),i(i(u,y),i(u,i(x,z))))).
79 [hyper,1,73,73] P(i(i(x,i(y,i(z,u))),i(z,i(x,i(y,u))))).
81 [hyper,1,73,71] P(i(i(x,i(y,z)),i(i(u,y),i(x,i(u,z))))).
85 [hyper,1,3,74] P(i(i(x,i(y,z)),i(x,i(i(z,u),i(y,u))))).
86 [hyper,1,74,4] P(i(i(i(x,i(y,z)),u),i(i(y,i(x,z)),u))).
87 [hyper,1,74,3] P(i(i(i(i(x,y),i(x,z)),u),i(i(y,z),u))).
90 [hyper,1,3,75] P(i(i(x,y),i(x,i(i(y,z),z)))).
92 [hyper,1,75,5] P(i(i(i(x,x),y),y)).
118 [hyper,1,85,4] P(i(i(x,i(y,z)),i(i(i(x,z),u),i(y,u)))).
133 [hyper,1,87,73] P(i(i(x,i(y,z)),i(i(u,x),i(y,i(u,z))))).
138 [hyper,1,85,90] P(i(i(x,y),i(i(i(i(y,z),z),u),i(x,u)))).
151 [hyper,1,81,92] P(i(i(x,y),i(i(i(z,z),i(y,u)),i(x,u)))).
155 [hyper,1,74,92] P(i(i(x,y),i(i(i(z,z),x),y))).
160 [hyper,1,87,118] P(i(i(x,y),i(i(i(i(z,x),y),u),i(z,u)))).
162 [hyper,1,118,92] P(i(i(i(i(i(x,x),i(y,z)),z),u),i(y,u))).
164 [hyper,1,87,133] P(i(i(x,y),i(i(z,i(u,x)),i(u,i(z,y))))).
175 [hyper,1,151,75] P(i(i(i(x,x),i(i(i(y,z),z),u)),i(y,u))).
180 [hyper,1,86,155] P(i(i(x,i(y,z)),i(i(i(u,u),y),i(x,z)))).
186 [hyper,1,74,155] P(i(i(i(i(i(x,x),y),z),u),i(i(y,z),u))).
190 [hyper,1,155,138] P(i(i(i(x,x),i(y,z)),i(i(i(i(z,u),u),v),i(y,v)))).
191 [hyper,1,155,133] P(i(i(i(x,x),i(y,i(z,u))),i(i(v,y),i(z,i(v,u))))).
198 [hyper,1,155,160] P(i(i(i(x,x),i(y,z)),i(i(i(i(u,y),z),v),i(u,v)))).
204 [hyper,1,85,162] P(i(i(i(i(i(x,x),i(y,z)),z),u),i(i(u,v),i(y,v)))).
212 [hyper,1,155,164] P(i(i(i(x,x),i(y,z)),i(i(u,i(v,y)),i(v,i(u,z))))).
218 [hyper,1,85,175] P(i(i(i(x,x),i(i(i(y,z),z),u)),i(i(u,v),i(y,v)))).
219 [hyper,1,81,175] P(i(i(x,y),i(i(i(z,z),i(i(i(y,u),u),v)),i(x,v)))).
225 [hyper,1,180,164] P(i(i(i(x,x),i(y,i(z,u))),i(i(u,v),i(z,i(y,v))))).
227 [hyper,1,180,133] P(i(i(i(x,x),i(y,z)),i(i(z,i(u,v)),i(u,i(y,v))))).
230 [hyper,1,180,77] P(i(i(i(x,x),i(y,z)),i(i(u,i(z,v)),i(y,i(u,v))))).
234 [hyper,1,186,118] P(i(i(x,i(y,z)),i(i(i(i(i(u,u),x),z),v),i(y,v)))).
237 [hyper,1,86,191] P(i(i(x,i(i(y,y),i(z,u))),i(i(v,x),i(z,i(v,u))))).
238 [hyper,1,79,191] P(i(x,i(i(i(y,y),i(z,i(x,u))),i(i(v,z),i(v,u))))).
239 [hyper,1,4,191] P(i(i(x,y),i(i(i(z,z),i(y,i(u,v))),i(u,i(x,v))))).
249 [hyper,1,4,212] P(i(i(x,i(y,z)),i(i(i(u,u),i(z,v)),i(y,i(x,v))))).
258 [hyper,1,218,219] P(i(i(i(x,y),z),i(i(i(i(x,u),u),y),z))).
262 [hyper,1,86,225] P(i(i(x,i(i(y,y),i(z,u))),i(i(u,v),i(z,i(x,v))))).
272 [hyper,1,86,227] P(i(i(x,i(i(y,y),z)),i(i(z,i(u,v)),i(u,i(x,v))))).
273 [hyper,1,79,227] P(i(x,i(i(i(y,y),i(z,u)),i(i(u,i(x,v)),i(z,v))))).
274 [hyper,1,73,227] P(i(i(i(x,x),i(y,z)),i(u,i(i(z,i(u,v)),i(y,v))))).
275 [hyper,1,4,227] P(i(i(x,i(y,z)),i(i(i(u,u),i(v,x)),i(y,i(v,z))))).
279 [hyper,1,4,230] P(i(i(x,i(y,z)),i(i(i(u,u),i(v,y)),i(v,i(x,z))))).
282 [hyper,1,4,237] P(i(i(x,y),i(i(y,i(i(z,z),i(u,v))),i(u,i(x,v))))).
286 [hyper,1,73,239] P(i(i(x,y),i(z,i(i(i(u,u),i(y,i(z,v))),i(x,v))))).
292 [hyper,1,79,249] P(i(x,i(i(y,i(x,z)),i(i(i(u,u),i(z,v)),i(y,v))))).
293 [hyper,1,73,249] P(i(i(x,i(y,z)),i(y,i(i(i(u,u),i(z,v)),i(x,v))))).
298 [hyper,1,258,151] P(i(i(i(i(x,y),y),z),i(i(i(u,u),i(z,v)),i(x,v)))).
301 [hyper,1,73,272] P(i(i(x,i(i(y,y),z)),i(u,i(i(z,i(u,v)),i(x,v))))).
308 [hyper,1,87,298] P(i(i(x,y),i(i(i(z,z),i(i(i(u,x),y),v)),i(u,v)))).
309 [hyper,1,4,298] P(i(i(i(x,x),i(y,z)),i(i(i(i(u,v),v),y),i(u,z)))).

You see that the fantasy goal was in fact realized: A proof of no more than fifty-six steps was eventually found, in fact, one of length fifty-three. That proof contains exactly twenty-nine formulas relying on five distinct variables, each of length nineteen (not counting the predicate or parentheses or commas). The proof might be interesting to study in detail in that its level is but 8 even though its length is 53, which makes it (in my experience) most unusual; indeed, proofs of roughly that length typically have much higher level. (The level of an input clause is 0, and the level of a deduced clause is one greater than the maximum of the levels of its parents.) Therefore, I made a conjecture. I wondered whether the sole item among the twenty-nine that is not among the twenty-eight could be a single axiom for the *BCI* logic. And you have the wellspring for Chapter 3.

4. Chapter 3 Beauty Beheld

Halleck immediately decided to show that the conjecture held, that what we called the 29th formula is a single axiom.

P(i(i(i(x,x),i(y,i(z,u))),i(i(u,v),i(z,i(y,v))))). % a 29th not among the 28 known singles for BCI.

After all, the easy half was in hand: The 29th had been shown to be a theorem of the *BCI* logic, as part of a 53-step proof that used as hypotheses *B*, *C*, and *I*. He chose to seek a proof that deduced each of the just-cited three axioms, with the 29th formula as sole hypothesis. He used his reasoning program shotgun.

To my delight, he succeeded. Indeed, he sent me a 30-step proof showing that the 29th formula is sufficiently powerful to permit the deduction of the 3-axiom system for *BCI* when condensed detachment is employed as the sole inference rule. The entire story, to this point, was begun and finished in less than three days of real time. The new single axiom is one of fifty-seven that Ulrich sent Halleck and that he had earlier shown were the only remaining 5-variable candidates of length 19 for being single axioms for this area of logic. Yes, requiring but three days for the entire investigation still brings me great satisfaction. Naturally, as you would predict, I turned to the seeking of a proof shorter than thirty, a proof that deduces the join of *B*, *C*, and *I* from formula 29.

As you will see from the input file, Input File 3, that I give almost immediately, I deviated from my usual approach. In particular, although I included as resonators the correspondents of the Halleck 30-step proof, I also included other resonators from earlier experiments. That latter inclusion resulted, I believe, from the fact that I simply turned to a database of input files and proofs and selected one that seemed appropriate. If you wish to have in hand the Halleck 30-step proof, perhaps you can take the cited thirty resonators and proceed accordingly. I just remind you that, when given a proof to shorten, I do not consider the history but am merely concerned with the deduced formulas or equations. Yes, what I have given in this notebook applies to problems in which equality is the only or dominant relation, problems, say, from various areas of abstract algebra.

Input File 3

```
set(hyper_res).
assign(max_weight,40).
% assign(change_limit_after,800).
% assign(new_max_weight,22).
assign(max_proofs,-1).
clear(print_kept).
set(ancestor_subsume).
set(back_sub).
% clear(for_sub).
clear(print_back_sub).
clear(print_kept).
clear(print_new_demod).
clear(print_back_demod).
clear(print_back_sub).
assign(max_distinct_vars,10).
assign(pick_given_ratio,4).
assign(max_mem,750000).
```

```
assign(report,5400).
set(order_history).
set(input_sos_first).
%  set(sos_queue).
assign(bsub_hint_wt,1).
set(keep_hint_subsumers).

weight_list(pick_and_purge).
%  Following 30 from Halleck purport to prove a new single ax for BCI, the 29th 5-var in a 53-step proof.
weight(P(i(i(i(i(x,y),z),u),i(x,i(i(y,z),u)))),-4).
weight(P(i(i(i(i(x,y),z),u),i(i(y,z),i(x,u)))),-4).
weight(P(i(i(x,y),i(i(z,u),i(i(u,x),i(z,y))))),-4).
weight(P(i(i(i(x,y),z),i(i(u,y),i(i(x,u),z)))),-4).
weight(P(i(i(x,y),i(i(z,u),i(z,i(i(u,x),y))))),-4).
weight(P(i(i(i(i(x,y),y),z),i(u,i(i(u,x),z)))),-4).
weight(P(i(x,i(i(x,i(y,z)),i(y,i(i(z,u),u))))),-4).
weight(P(i(i(i(i(x,y),y),z),i(u,i(i(i(v,v),i(u,x)),z)))),-4).
weight(P(i(i(i(i(x,y),y),z),i(i(i(u,u),i(v,x)),i(v,z)))),-4).
weight(P(i(i(i(x,x),i(y,i(z,u))),i(y,i(i(u,v),i(z,v))))),-4).
weight(P(i(i(x,y),i(i(i(i(y,z),z),u),i(x,u)))),-4).
weight(P(i(i(i(i(x,y),y),z),i(i(z,u),i(x,u)))),-4).
weight(P(i(i(i(i(x,y),i(i(z,x),y)),u),i(z,u))),-4).
weight(P(i(x,i(i(x,i(y,z)),i(u,i(i(u,y),z))))),-4).
weight(P(i(i(i(x,x),i(y,z)),i(i(i(i(u,y),z),v),i(u,v)))),-4).
weight(P(i(i(i(i(x,i(i(i(y,z),z),u)),i(y,u)),v),i(x,v))),-4).
weight(P(i(i(x,y),i(i(z,x),i(z,y)))),-4).
weight(P(i(i(x,y),i(i(y,z),i(x,z)))),-4).
weight(P(i(i(i(i(x,i(y,z)),i(y,z)),u),i(x,u))),-4).
weight(P(i(i(x,i(i(i(y,z),z),u)),i(x,i(y,u)))),-4).
weight(P(i(i(x,i(y,z)),i(y,i(x,z)))),-4).
weight(P(i(i(x,y),i(z,i(i(z,x),y)))),-4).
weight(P(i(i(i(x,x),i(y,z)),i(i(i(i(z,u),u),v),i(y,v)))),-4).
weight(P(i(i(i(i(i(i(x,y),y),z),z),u),i(x,u))),-4).
weight(P(i(i(x,y),i(x,y))),-4).
weight(P(i(i(i(i(x,y),y),z),i(x,z))),-4).
weight(P(i(x,i(i(x,y),y))),-4).
weight(P(i(i(i(x,x),i(y,z)),i(i(u,y),i(u,z)))),-4).
weight(P(i(i(x,i(i(y,y),z)),i(x,z))),-4).
weight(P(i(x,x)),-4).
%  Following 31 include Meredith's single for BCK, and his 30-step proof.
weight(P(i(i(i(u,v),w),i(i(x,i(w,y)),i(v,i(x,y))))),2).  %  Meredith's single axiom for BCK
weight(P(i(i(u,i(i(i(v,i(w,x)),i(y,i(v,x))),z)),i(w,i(u,z)))),2).
weight(P(i(u,i(i(i(v,w),x),i(w,i(i(y,i(u,z)),i(y,z)))))),2).
weight(P(i(u,i(v,i(i(u,w),i(i(x,i(v,y)),i(x,y)))))),2).
weight(P(i(u,i(i(i(i(i(v,w),x),i(i(y,i(x,z)),i(w,i(y,z)))),v6),i(i(v7,i(u,v8)),i(v7,v8))))),2).
weight(P(i(i(u,i(v,w)),i(x,i(i(y,i(x,z)),i(y,z))))),2).
weight(P(i(u,i(i(v,i(u,w)),i(v,w)))),2).
weight(P(i(i(u,i(i(v,i(i(w,i(v,x)),i(w,x))),y)),i(u,y))),2).
weight(P(i(i(u,i(v,w)),i(v,i(u,w)))),2).  %  C
weight(P(i(i(i(u,i(i(v,i(u,w)),i(v,w))),i(i(x,i(i(y,i(x,z)),i(y,z))),v6)),v6)),2).
weight(P(i(i(u,i(v,w)),i(i(i(x,i(i(y,i(x,z)),i(y,z))),v),i(u,w)))),2).
weight(P(i(i(u,i(i(i(v,i(i(w,i(v,x)),i(w,x))),y),i(z,v6)),v7)),i(i(y,v6),i(u,v7)))),2).
weight(P(i(i(u,v),i(w,i(i(i(x,i(i(y,i(x,z)),i(y,z))),u),v)))),2).
```

```
weight(P(i(i(u,v),i(i(i(w,i(i(x,i(w,y)),i(x,y))),u),v))),2).
weight(P(i(i(i(u,i(i(v,i(u,w)),i(v,w))),i(x,y)),i(i(z,i(i(v6,i(z,v7)),i(v6,v7))),x),y))),2).
weight(P(i(i(u,i(i(i(i(v,i(i(w,i(v,x)),i(w,x))),y),z),v6)),i(i(y,z),i(u,v6)))),2).
weight(P(i(i(u,i(i(v,i(w,x)),y)),i(i(v,x),i(u,y)))),2).
weight(P(i(i(u,v),i(w,i(u,v)))),2).
weight(P(i(i(u,i(i(v,i(w,x)),y)),i(x,i(u,y)))),2).
weight(P(i(u,i(v,i(w,u)))),2).  %  K
weight(P(i(u,i(v,u))),2). % K
weight(P(i(u,i(i(i(v,w),x),i(i(y,i(x,z)),i(w,i(y,z)))))),2).
weight(P(i(i(u,v),i(w,i(i(x,i(v,y)),i(u,i(x,y)))))),2).
weight(P(i(i(u,v),i(i(w,i(v,x)),i(u,i(w,x))))),2).
weight(P(i(i(u,i(v,w)),i(i(x,v),i(x,i(u,w))))),2).
weight(P(i(i(u,v),i(u,v))),2).
weight(P(i(i(u,v),i(u,i(i(v,w),w)))),2).
weight(P(i(i(u,v),i(i(i(u,i(i(v,w),w)),x),x))),2).
weight(P(i(i(i(u,i(i(v,w),w)),x),i(i(u,v),x))),2).  %  B'
weight(P(i(i(u,v),i(i(v,w),i(u,w)))),2).  %  B'
weight(P(i(i(u,v),i(i(w,u),i(w,v)))),2).  %  B
end_of_list.

list(usable).
-P(i(x,y)) | -P(x) | P(y).
-P(i(i(a1,i(b,a2)),i(b,i(a1,a2)))) | -P(i(a1,a1)) | -P(i(i(a1,b),i(i(a2,a1),i(a2,b)))) |
    $ANS(all).  % BCI
end_of_list.

list(sos).
P(i(i(i(x,x),i(y,i(z,u))),i(i(u,v),i(z,i(y,v))))).  %  a 29th not among the
%     28 known singles for BCI.
%  P(i(i(x,i(y,z)),i(i(i(u,u),i(v,y)),i(v,i(x,z))))).  % M's BCI #1
end_of_list.

list(passive).
%  Following is neg of I, of BCI.
-P(i(a,a)) | $ANS(I).
%  Following 30 prove B,C,K from Meredith single.
-P(i(i(a1,i(b,a2)),i(b,i(a1,a2)))) | $ANS(C).  %  C
-P(i(a1,i(b,a1))). % K
-P(i(i(a1,b),i(i(b,a2),i(a1,a2)))) | $ANS(BP).  %  B'
-P(i(i(a1,b),i(i(a2,a1),i(a2,b)))) | $ANS(B).  % B
end_of_list.
```

The submission to OTTER of Input File 3 yielded, respectively, proofs of length 34, 33, 32, 31, 32, 32, 28, 27, 32, 31, 25, 23, 27, 25, and 27.

I continued my search for an even shorter proof than length 23, applying the usual approach, sometimes leading to an uninteresting result. The first of the fifteen proofs was completed in approximately 2 CPU-seconds, and the last was completed in approximately 40500 CPU-seconds. The levels range from 18 to 20. Finally, I had a 21-step level-16 proof, the following, which I invite you to consider obtaining on your own.

A Satisfying 21-Step Proof

```
----- Otter 3.3g-work, Jan 2005 -----
```

The process was started by wos on octopus.mcs.anl.gov,
Fri Feb 20 13:43:10 2009
The command was "otter". The process ID is 13968.

-----> EMPTY CLAUSE at 0.16 sec ----> 2472 [hyper,2,1042,2449,1280] $ANS(all).

Length of proof is 21. Level of proof is 16.

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

1 [] -P(i(x,y)) | -P(x) | P(y).
2 [] -P(i(i(a1,i(b,a2)),i(b,i(a1,a2)))) | -P(i(a1,a1)) | -P(i(i(a1,b),i(i(a2,a1),i(a2,b)))) |
$ANS(all).
3 [] P(i(i(i(x,x),i(y,i(z,u))),i(i(u,v),i(z,i(y,v))))).
13 [hyper,1,3,3] P(i(i(i(i(x,y),z),u),i(x,i(i(y,z),u)))).
15 [hyper,1,3,13] P(i(i(i(i(x,y),z),u),i(i(y,z),i(x,u)))).
25 [hyper,1,15,13] P(i(i(x,y),i(i(z,u),i(z,i(i(u,x),y))))).
31 [hyper,1,3,25] P(i(i(i(i(x,y),y),z),i(u,i(i(u,x),z)))).
47 [hyper,1,31,31] P(i(x,i(i(x,i(y,z)),i(u,i(i(u,y),z))))).
58 [hyper,1,31,13] P(i(x,i(i(x,i(y,z)),i(y,i(i(z,u),u))))).
162 [hyper,1,3,58] P(i(i(i(i(x,y),y),z),i(u,i(i(i(v,v),i(u,x)),z)))).
216 [hyper,1,3,162] P(i(i(i(i(x,y),y),z),i(i(i(u,u),i(v,x)),i(v,z)))).
323 [hyper,1,216,15] P(i(i(i(x,x),i(y,i(z,u))),i(y,i(i(u,v),i(z,v))))).
368 [hyper,1,323,216] P(i(i(i(x,x),i(y,z)),i(i(i(i(z,u),u),v),i(y,v)))).
374 [hyper,1,323,47] P(i(i(i(x,x),i(y,z)),i(i(i(i(u,y),z),v),i(u,v)))).
376 [hyper,1,323,25] P(i(i(x,y),i(i(i(i(y,z),z),u),i(x,u)))).
640 [hyper,1,374,374] P(i(i(i(i(x,i(i(i(y,z),z),u)),i(y,u)),v),i(x,v))).
889 [hyper,1,640,640] P(i(i(x,i(i(i(i(i(y,z),z),u),u),v)),i(x,i(y,v)))).
1042 [hyper,1,889,376] P(i(i(x,i(y,z)),i(y,i(x,z)))).
1045 [hyper,1,889,368] P(i(i(i(x,x),i(y,i(z,u))),i(z,i(y,u)))).
1280 [hyper,1,3,1042] P(i(i(x,y),i(i(z,x),i(z,y)))).
1336 [hyper,1,1045,47] P(i(x,i(i(i(y,y),i(z,u)),i(i(x,z),u)))).
2137 [hyper,1,1045,1336] P(i(i(i(x,x),y),i(i(i(z,z),i(y,u)),u))).
2377 [hyper,1,2137,1045] P(i(i(i(x,x),i(i(y,i(i(y,z),z)),u)),u)).
2449 [hyper,1,2377,2137] P(i(x,x)).

You might enjoy knowing that, rather than seeking a proof of the join of *B*, *C*, and *I* when studying a possible new single axiom for the logic, Ulrich generally prefers as targets the various known single axioms. I believe it accurate to say that he says the sought-after proof is typically easier to complete, when compared with seeking a proof of the join. In emulation of Ulrich, I today (March 1, 2009) sought such a proof from the 29th formula. The best I have so far has length and level equal to 9, the following proof. (Halleck had found an almost-identical proof as a subproof of a longer proof he obtained, pointed out by Ulrich.) You might seek a proof shorter than length nine that uses the 29th formula as hypothesis and condensed detachment to deduce one of various single axioms for *BCI*.

A 9-Step Proof Deducing from the 29th Formula a Known Single Axiom

----- Otter 3.3g-work, Jan 2005 -----
The process was started by wos on octopus.mcs.anl.gov,
Sun Mar 1 18:36:11 2009
The command was "otter". The process ID is 672.

----> UNIT CONFLICT at 0.02 sec ----> 299 [binary,298.1,14.1] $ANS(TARG).

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

```
---------------- PROOF ----------------

1 [] -P(i(x,y)) | -P(x) | P(y).
3 [] P(i(i(i(x,x),i(y,i(z,u))),i(i(u,v),i(z,i(y,v))))).
1 14 [] -P(i(i(i(a1,a1),i(a2,a3)),i(i(i(a3,a4),a4),a5),i(a2,a5)))) | $ANS(TARG).
39 [hyper,1,3,3] P(i(i(i(i(x,y),z),u),i(x,i(i(y,z),u)))).
41 [hyper,1,3,39] P(i(i(i(i(x,y),z),u),i(i(y,z),i(x,u)))).
51 [hyper,1,41,39] P(i(i(x,y),i(i(z,u),i(z,i(i(u,x),y))))).
57 [hyper,1,3,51] P(i(i(i(i(x,y),y),z),i(u,i(i(u,x),z)))).
77 [hyper,1,57,39] P(i(x,i(i(x,i(y,z)),i(y,i(i(z,u),u))))).
155 [hyper,1,3,77] P(i(i(i(i(x,y),y),z),i(u,i(i(i(v,v),i(u,x)),z)))).
196 [hyper,1,3,155] P(i(i(i(i(x,y),y),z),i(i(i(u,u),i(v,x)),i(v,z)))).
263 [hyper,1,196,41] P(i(i(i(x,x),i(y,i(z,u))),i(y,i(i(u,v),i(z,v))))).
298 [hyper,1,263,196] P(i(i(i(x,x),i(y,z)),i(i(i(i(z,u),u),v),i(y,v)))).
```

I was most pleased with learning of the Halleck proof that formula 29, taken as the sole hypothesis, suffices to deduce the 3-axiom system prominent in this notebook, thus showing it to be a single axiom. I was quite soon curious about whether a method I had employed in various notebooks would have enabled OTTER to find a proof of that fact, a method that is termed the *subformula strategy*. The essence of the strategy has you take the hypotheses, formula 29, and the targets, in this case the three axioms B, C, and I, and focus on their subformulas. You ignore the subformulas that are each just a variable, but you do take each of the entire formulas as part of the set in focus. I submitted the following input file for consideration.

Input File 4

```
set(hyper_res).
assign(max_weight,19).
assign(change_limit_after,400).
assign(new_max_weight,8).
assign(max_proofs,-1).
clear(print_kept).
%  set(ancestor_subsume).
set(back_sub).
% clear(for_sub).
clear(print_back_sub).
clear(print_kept).
clear(print_new_demod).
clear(print_back_demod).
clear(print_back_sub).
assign(max_distinct_vars,6).
%  assign(pick_given_ratio,4).
assign(max_mem,750000).
assign(report,5400).
set(order_history).
set(input_sos_first).
set(sos_queue).
assign(bsub_hint_wt,1).
set(keep_hint_subsumers).

weight_list(pick_and_purge).
%  Following 19 are interesting subformulas of the hyp and of BCI.
```

```
weight(i(u,v),1).
weight(i(y,i(z,u)),1).
weight(i(i(u,v),i(z,i(y,v))),1).
weight(i(i(i(x,x),i(y,i(z,u))),i(i(u,v),i(z,i(y,v)))),1).
weight(P(i(i(i(x,x),i(y,i(z,u))),i(i(u,v),i(z,i(y,v))))),1).
weight(i(x,x),1).
weight(P(i(x,x)),1).
weight(i(x,z),1).
weight(i(y,z),1).
weight(i(x,i(y,z)),1).
weight(i(y,i(x,z)),1).
weight(i(i(x,i(y,z)),i(y,i(x,z))),1).
weight(P(i(i(x,i(y,z)),i(y,i(x,z)))),1).
weight(i(x,y),1).
weight(i(z,x),1).
weight(i(z,y),1).
weight(i(i(z,x),i(z,y)),1).
weight(i(i(x,y),i(i(z,x),i(z,y))),1).
weight(P(i(i(x,y),i(i(z,x),i(z,y)))),1).
end_of_list.

list(usable).
-P(i(x,y)) | -P(x) | P(y).
-P(i(i(a1,i(b,a2)),i(b,i(a1,a2)))) | -P(i(a1,a1)) | -P(i(i(a1,b),i(i(a2,a1),i(a2,b)))) |
    $ANS(all).  % BCI
end_of_list.

list(sos).
P(i(i(i(x,x),i(y,i(z,u))),i(i(u,v),i(z,i(y,v))))).  %  a 29th not among the 28 known singles for BCI.
%  P(i(i(x,i(y,z)),i(i(i(u,u),i(v,y)),i(v,i(x,z))))).  % M's BCI #1
end_of_list.

list(passive).
-P(i(a,a)) | $ANS(I).
-P(i(i(a1,i(b,a2)),i(b,i(a1,a2)))) | $ANS(C).  %  C
-P(i(i(a1,b),i(i(a2,a1),i(a2,b)))) | $ANS(B).  % B
end_of_list.
```

The approach succeeded, yielding the following proof, far longer than Halleck's original 30-step proof.

A Proof Obtained with the Subformula Strategy

```
----- Otter 3.3g-work, Jan 2005 -----
The process was started by wos on elephant.mcs.anl.gov,
Thu Feb 19 16:31:07 2009
The command was "otter".  The process ID is 13840.

-----> EMPTY CLAUSE at 1670.60 sec ----> 8858 [hyper,2,8080,8801,7152] $ANS(all).

Length of proof is 82.  Level of proof is 13.

---------------- PROOF ----------------
```

1 [] -P(i(x,y)) | -P(x) | P(y).
2 [] -P(i(i(a1,i(b,a2)),i(b,i(a1,a2)))) | -P(i(a1,a1)) | -P(i(i(a1,b),i(i(a2,a1),i(a2,b)))) | $ANS(all).
3 [] P(i(i(i(x,x),i(y,i(z,u))),i(i(u,v),i(z,i(y,v))))).
7 [hyper,1,3,3] P(i(i(i(i(x,y),z),u),i(x,i(i(y,z),u)))).
9 [hyper,1,3,7] P(i(i(i(i(x,y),z),u),i(i(y,z),i(x,u)))).
10 [hyper,1,7,3] P(i(x,i(i(x,i(y,i(z,u))),i(i(u,v),i(z,i(y,v)))))).
14 [hyper,1,9,9] P(i(i(x,y),i(i(z,u),i(i(u,x),i(z,y))))).
16 [hyper,1,7,9] P(i(i(x,y),i(i(z,u),i(i(y,z),i(x,u))))).
18 [hyper,1,9,7] P(i(i(x,y),i(i(z,u),i(z,i(i(u,x),y))))).
19 [hyper,1,9,3] P(i(i(x,i(y,i(z,u))),i(x,i(i(u,v),i(z,i(y,v)))))).
20 [hyper,1,3,10] P(i(i(i(x,i(y,z)),u),i(i(v,z),i(i(i(w,w),i(y,i(x,v))),u)))).
25 [hyper,1,9,14] P(i(i(x,y),i(z,i(i(u,v),i(i(v,i(z,x)),i(u,y)))))).
28 [hyper,1,3,14] P(i(i(i(x,y),z),i(i(u,y),i(i(x,u),z)))).
38 [hyper,1,3,16] P(i(i(i(x,y),z),i(i(x,u),i(i(u,y),z)))).
49 [hyper,1,3,18] P(i(i(i(i(x,y),y),z),i(u,i(i(u,x),z)))).
62 [hyper,1,19,14] P(i(i(x,y),i(i(i(z,y),u),i(i(v,x),i(i(z,v),u))))).
69 [hyper,1,9,20] P(i(i(i(x,y),z),i(u,i(i(v,y),i(i(i(w,w),i(x,i(u,v))),z))))).
82 [hyper,1,3,25] P(i(i(i(i(x,i(y,z)),i(u,z)),v),i(i(u,x),i(y,v)))).
103 [hyper,1,28,3] P(i(i(x,i(y,i(z,u))),i(i(i(v,v),x),i(i(u,w),i(z,i(y,w)))))).
161 [hyper,1,49,49] P(i(x,i(i(x,i(y,z)),i(u,i(i(u,y),z))))).
171 [hyper,1,3,49] P(i(i(i(i(x,y),y),z),i(i(u,x),i(u,z)))).
174 [hyper,1,49,38] P(i(x,i(i(x,y),i(i(y,z),i(i(z,u),u))))).
176 [hyper,1,49,28] P(i(x,i(i(x,y),i(i(z,u),i(i(y,z),u))))).
185 [hyper,1,49,9] P(i(x,i(i(x,i(y,z)),i(i(z,u),i(y,u))))).
187 [hyper,1,49,7] P(i(x,i(i(x,i(y,z)),i(y,i(i(z,u),u))))).
253 [hyper,1,3,62] P(i(i(i(i(x,y),z),u),i(i(y,v),i(i(i(x,v),z),u)))).
314 [hyper,1,9,69] P(i(i(x,y),i(z,i(u,i(i(v,x),i(i(i(w,w),i(z,i(u,v))),y)))))).
316 [hyper,1,3,69] P(i(i(i(i(i(x,x),i(y,i(z,u))),i(y,v)),w),i(i(u,v),i(z,w)))).
377 [hyper,1,49,82] P(i(x,i(i(x,i(y,i(z,u))),i(i(v,y),i(z,i(v,u)))))).
398 [hyper,1,82,20] P(i(i(x,y),i(z,i(i(u,v),i(i(i(w,w),i(z,i(y,u))),i(x,v)))))).
546 [hyper,1,103,49] P(i(i(i(x,x),i(i(i(y,z),z),u)),i(i(u,v),i(i(w,y),i(w,v))))).
974 [hyper,1,3,161] P(i(i(i(i(x,y),z),u),i(x,i(i(i(v,v),i(y,z)),u)))).
1112 [hyper,1,171,171] P(i(i(x,i(y,z)),i(x,i(i(u,y),i(u,z))))).
1217 [hyper,1,171,7] P(i(i(x,i(y,z)),i(x,i(y,i(i(z,u),u))))).
1240 [hyper,1,3,174] P(i(i(i(i(x,y),y),z),i(i(u,x),i(i(i(v,v),u),z)))).
1258 [hyper,1,3,176] P(i(i(i(i(x,y),z),u),i(i(y,z),i(i(i(v,v),x),u)))).
1285 [hyper,1,3,185] P(i(i(i(x,y),z),i(i(u,y),i(i(i(v,v),i(x,u)),z)))).
1307 [hyper,1,3,187] P(i(i(i(i(x,y),y),z),i(u,i(i(i(v,v),i(u,x)),z)))).
1681 [hyper,1,3,253] P(i(i(i(i(x,y),z),u),i(i(i(x,v),z),i(i(y,v),u)))).
2403 [hyper,1,3,314] P(i(i(i(i(x,y),i(i(i(z,z),i(u,i(v,x))),y)),w),i(v,i(u,w)))).
2405 [hyper,1,171,316] P(i(i(x,i(i(y,y),i(z,i(u,v)))),i(x,i(i(v,w),i(u,i(z,w)))))).
2849 [hyper,1,3,377] P(i(i(i(x,i(y,z)),u),i(i(y,v),i(i(i(w,w),i(v,i(x,z))),u)))).
3365 [hyper,1,3,398] P(i(i(i(i(i(x,x),i(y,i(z,u))),i(z,v)),w),i(i(u,v),i(y,w)))).
3779 [hyper,1,316,1112] P(i(i(x,y),i(z,i(i(i(u,u),i(v,i(z,x))),i(i(w,v),i(w,y)))))).
4049 [hyper,1,1240,9] P(i(i(x,i(y,z)),i(i(i(u,u),x),i(i(z,v),i(y,v))))).
4054 [hyper,1,1258,171] P(i(i(x,y),i(i(i(z,z),i(u,x)),i(i(v,u),i(v,y))))).
4086 [hyper,1,546,1285] P(i(i(i(i(i(x,x),i(y,i(z,u))),i(y,u)),v),i(i(w,z),i(w,v)))).
4103 [hyper,1,1307,171] P(i(x,i(i(i(y,y),i(x,i(z,u))),i(i(v,z),i(v,u))))).
4104 [hyper,1,1307,9] P(i(x,i(i(i(y,y),i(x,i(z,u))),i(i(u,v),i(z,v))))).
4460 [hyper,1,2405,171] P(i(i(i(i(x,y),y),i(z,u)),i(i(u,v),i(z,i(x,v))))).
4461 [hyper,1,2405,3] P(i(i(i(x,x),i(y,i(z,u))),i(i(u,v),i(y,i(z,v))))).
5249 [hyper,1,3,3779] P(i(i(i(i(x,y),i(x,z)),u),i(i(i(v,v),i(y,i(w,z))),i(w,u)))).

```
5499 [hyper,1,4049,1112] P(i(i(i(x,x),i(y,i(z,u))),i(i(i(i(v,z),i(v,u)),w),i(y,w)))).
5511 [hyper,1,171,4086] P(i(i(x,i(i(y,y),i(z,i(u,v)))),i(x,i(i(w,u),i(w,i(z,v)))))).
5515 [hyper,1,4086,1681] P(i(i(x,y),i(x,i(i(i(z,u),i(v,i(y,w))),i(i(z,u),i(v,w)))))).
5640 [hyper,1,2405,4460] P(i(i(i(i(x,y),y),i(z,u)),i(i(u,v),i(x,i(z,v))))).
5648 [hyper,1,4460,2849] P(i(i(i(i(i(x,x),i(y,i(z,u))),i(v,u)),w),i(i(v,y),i(z,w)))).
5649 [hyper,1,4460,1285] P(i(i(i(i(i(x,x),i(y,z)),u),v),i(i(z,u),i(y,v)))).
5992 [hyper,1,5499,4054] P(i(i(i(i(x,i(y,z)),i(x,i(y,u))),v),i(i(i(w,w),i(z,u)),v))).
5997 [hyper,1,5511,5249] P(i(i(i(i(x,i(y,z)),i(x,z)),i(u,v)),i(i(w,u),i(w,i(y,v))))).
6003 [hyper,1,5511,4461] P(i(i(i(x,x),i(y,i(z,u))),i(i(v,z),i(v,i(y,u))))).
6010 [hyper,1,5511,1217] P(i(i(i(x,x),i(y,z)),i(i(u,i(z,v)),i(u,i(y,v))))).
6015 [hyper,1,5511,3] P(i(i(i(x,x),i(y,i(z,u))),i(i(v,y),i(v,i(z,u))))).
6018 [hyper,1,3,5515] P(i(i(i(i(x,y),i(z,u)),v),i(i(i(x,y),i(z,i(w,u))),i(w,v)))).
6089 [hyper,1,5640,974] P(i(i(i(i(i(x,x),i(y,z)),z),u),i(i(v,y),i(v,u)))).
6141 [hyper,1,171,5648] P(i(i(x,i(i(y,y),i(z,i(u,v)))),i(x,i(i(w,z),i(u,i(w,v)))))).
6158 [hyper,1,5511,5649] P(i(i(i(i(i(x,x),i(y,z)),z),i(u,v)),i(i(w,u),i(w,i(y,v))))).
6370 [hyper,1,3365,5992] P(i(i(x,y),i(i(z,u),i(i(i(v,v),i(u,x)),i(z,y))))).
6443 [hyper,1,5649,5997] P(i(i(x,i(y,z)),i(i(u,x),i(i(v,y),i(v,i(u,z)))))).
6487 [hyper,1,5499,6010] P(i(i(i(i(x,y),i(x,i(z,u))),v),i(i(y,i(z,u)),v))).
6530 [hyper,1,6015,4104] P(i(i(x,i(i(y,y),i(i(z,z),i(u,v)))),i(x,i(i(v,w),i(u,w))))).
6531 [hyper,1,6015,4103] P(i(i(x,i(i(y,y),i(i(z,z),i(u,v)))),i(x,i(i(w,u),i(w,v))))).
6532 [hyper,1,6015,4054] P(i(i(x,i(i(y,y),i(z,u))),i(x,i(i(v,z),i(v,u))))).
6595 [hyper,1,6089,2403] P(i(i(x,i(i(y,y),i(z,i(u,v)))),i(x,i(u,i(z,v))))).
6717 [hyper,1,6158,6018] P(i(i(x,i(i(y,y),i(z,i(u,v)))),i(x,i(z,i(u,v))))).
7016 [hyper,1,6487,6003] P(i(i(x,i(y,z)),i(i(u,y),i(u,i(x,z))))).
7124 [hyper,1,6530,6443] P(i(i(x,i(y,z)),i(i(i(x,z),u),i(y,u)))).
7133 [hyper,1,6531,6531] P(i(i(i(x,x),i(i(y,y),i(i(z,z),i(u,v)))),i(i(w,u),i(w,v)))).
7152 [hyper,1,6531,6370] P(i(i(x,y),i(i(z,x),i(z,y)))).
7169 [hyper,1,6141,6532] P(i(i(i(x,x),i(i(y,y),i(z,u))),i(i(v,i(w,z)),i(w,i(v,u))))).
7321 [hyper,1,6595,6595] P(i(i(i(x,x),i(i(y,y),i(z,i(u,v)))),i(z,i(u,v)))).
8080 [hyper,1,6717,7016] P(i(i(x,i(y,z)),i(y,i(x,z)))).
8149 [hyper,1,7133,7124] P(i(i(x,i(i(y,y),z)),i(x,z))).
8236 [hyper,1,7321,7169] P(i(x,i(i(x,y),y))).
8801 [hyper,1,8149,8236] P(i(x,x)).
```

Far more important to me than the CPU time, more than 1670 CPU-seconds, required to complete a proof was the fact that the subformula strategy did lead to a proof, providing a bit more evidence of the possible value of this approach. You now have witnessed two approaches to proving a theorem, use of Halleck's program shotgun, and use of the subformula strategy. The goal of proving one theorem after another requires access to a variety of approaches.

Curiosity persisted. I wondered how short a proof I could find that used the 3-axiom system as hypothesis and sought to prove formula 29 alone. The following input file leads to an 8-step proof, which I also give.

Input File 5

```
set(hyper_res).
assign(max_weight,29).
% assign(max_seconds,6).
assign(change_limit_after,800).
assign(new_max_weight,22).
assign(max_proofs,-1).
clear(print_kept).
% set(process_input).
```

```
set(ancestor_subsume).
set(back_sub).
% clear(for_sub).
clear(print_back_sub).
clear(print_kept).
clear(print_new_demod).
clear(print_back_demod).
clear(print_back_sub).
% clear(print_given).
assign(max_distinct_vars,6).
%  assign(pick_given_ratio,2).
assign(max_mem,750000).
assign(report,5400).
set(order_history).
set(input_sos_first).
set(sos_queue).
% set(sos_stack).
assign(bsub_hint_wt,1).
set(keep_hint_subsumers).
assign(heat,0).
% assign(dynamic_heat_weight,2).

weight_list(pick_and_purge).
weight(P(i(i(x,y),i(z,i(i(z,x),y)))),1000).
weight(junk,1000).
end_of_list.

list(usable).
-P(i(x,y)) | -P(x) | P(y).
%  -P(i(i(a1,i(a2,a3)),i(i(i(a4,a4),i(a5,a2)),i(a5,i(a1,a3))))) | -P(i(i(i(a1,a1),i(a2,a3)),
    i(i(a4,i(a3,a5)),i(a2,i(a4,a5))))) | -P(i(i(a1,a2),i(i(i(a3,a3),i(i(i(a2,a4),a4),a5)),i(a1,a5)))) |
    -P(i(i(a1,a2),i(i(a2,i(i(a3,a3),i(a4,a5))),i(a4,i(a1,a5))))) | -P(i(i(i(i(i(a1,a1),
    i(a2,a3)),a3),a4),i(i(a4,a5),i(a2,a5)))) | -P(i(i(i(i(a1,a2),a2),a3),i(i(i(a4,a4),
    i(a3,a5)),i(a1,a5)))) | -P(i(i(a1,i(a2,a3)),i(i(i(i(i(a4,a4),a1),a3),a5),i(a2,a5)))) |
    -P(i(i(a1,i(a2,a3)),i(i(i(a4,a4),i(a3,a5)),i(a2,i(a1,a5))))) | -P(i(i(i(a1,a1),
    i(i(i(a2,a3),a3),a4)),i(i(a4,a5),i(a2,a5)))) | -P(i(i(i(a1,a1),i(a2,a3)),i(a4,i(i(a3,i(a4,a5)),
    i(a2,a5))))) | -P(i(i(i(a1,a1),i(a2,a3)),i(i(i(i(a3,a4),a4),a5),i(a2,a5)))) | -P(i(i(i(a1,a1),
    i(a2,a3)),i(i(a3,i(a4,a5)),i(a4,i(a2,a5))))) | -P(i(i(a1,i(i(a2,a2),i(a3,a4))),i(i(a4,a5),
    i(a3,i(a1,a5))))) | -P(i(i(a1,i(i(a2,a2),a3)),i(a4,i(i(a3,i(a4,a5)),i(a1,a5))))) |
    -P(i(i(a1,a2),i(i(i(a3,a3),i(a2,i(a4,a5))),i(a4,i(a1,a5))))) | -P(i(i(a1,i(a2,a3)),
    i(a2,i(i(i(a4,a4),i(a3,a5)),i(a1,a5))))) | -P(i(a1,i(i(i(a2,a2),i(a3,a4)),i(i(a4,i(a1,a5)),
    i(a3,a5))))) | -P(i(a1,i(i(a2,i(a1,a3)),i(i(i(a4,a4),i(a3,a5)),i(a2,a5))))) | -P(i(i(a1,a2),
    i(a3,i(i(i(a4,a4),i(a2,i(a3,a5))),i(a1,a5))))) | -P(i(i(i(a1,a1),i(a2,a3)),i(i(i(i(a4,a5),a5),a2),
    i(a4,a3)))) | -P(i(a1,i(i(i(a2,a2),i(a3,i(a1,a4))),i(i(a5,a3),i(a5,a4))))) | -P(i(i(i(a1,a1),
    i(a2,i(a3,a4))),i(i(a5,a2),i(a3,i(a5,a4))))) | -P(i(i(a1,a2),i(i(i(a3,a3),i(i(i(a4,a1),a2),a5)),
    i(a4,a5)))) | -P(i(i(a1,i(i(a2,a2),i(a3,a4))),i(i(a5,a1),i(a3,i(a5,a4))))) | -P(i(i(i(a1,a1),
    i(a2,a3)),i(i(i(i(a4,a2),a3),a5),i(a4,a5)))) | -P(i(i(i(a1,a1),i(a2,a3)),i(i(a4,i(a5,a2)),
    i(a5,i(a4,a3))))) | -P(i(i(a1,i(i(a2,a2),a3)),i(i(a3,i(a4,a5)),i(a4,i(a1,a5))))) |
    -P(i(i(a1,i(a2,a3)),i(i(i(a4,a4),i(a5,a1)),i(a2,i(a5,a3))))) | $ANS(TARGALL).
%  -P(i(i(a1,i(b,a2)),i(b,i(a1,a2)))) | -P(i(a1,a1)) | -P(i(i(a1,b),i(i(a2,a1),i(a2,b)))) |
     $ANS(all).  % BCI
end_of_list.
```

```
list(sos).
P(i(i(x,y),i(i(z,x),i(z,y)))).  % B
P(i(i(x,i(y,z)),i(y,i(x,z)))).  % C
P(i(x,x)).  % I
end_of_list.

list(passive).
%  Following is neg of the newest single found by Halleck and Wos, the 29th.
-P(i(i(i(a1,a1),i(a2,i(a3,a4))),i(i(a4,a5),i(a3,i(a2,a5))))) | $ANS(TARG29).  %  a 29th not among
%  the 28 known singles for BCI.
end_of_list.

%  Following purges unwanted formulas.
list(demodulators).
(P(i(i(i(i(x,y),y),z),i(x,z))) = junk).
(P(i(i(x,y),i(i(z,i(y,u)),i(x,i(z,u))))) = junk).
(P(i(i(x,i(y,z)),i(x,i(i(i(u,u),i(z,v)),i(y,v))))) = junk).
(P(i(i(x,y),i(i(y,i(z,u)),i(z,i(x,u))))) = junk).
%  (P(i(i(i(x,y),z),i(i(x,i(u,y)),i(u,z)))) = junk).
%  (P(i(i(i(x,x),i(y,z)),i(i(u,y),i(u,z)))) = junk).
%  (P(i(x,i(i(i(y,y),i(x,z)),z))) = junk).
%  (P(i(i(x,y),i(z,i(i(z,x),y)))) = junk).
%  (P(i(i(x,y),i(i(i(i(z,u),u),x),i(z,y)))) = junk).
%  (P(i(i(x,i(i(y,y),i(z,u))),i(z,i(x,u)))) = junk).
%  (P(i(i(x,i(i(y,y),z)),i(x,z))) = junk).
%  (P(i(i(x,i(y,z)),i(i(z,u),i(y,i(x,u))))) = junk).
%  (P(i(i(x,y),i(i(i(z,z),i(u,x)),i(u,y)))) = 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.

list(hints).
%  Following 58 purport to prove the join of the 28 singles for BCI.
P(i(i(x,i(y,z)),i(x,i(i(u,y),i(u,z))))).
P(i(i(x,i(y,i(z,u))),i(x,i(z,i(y,u))))).
P(i(i(x,y),i(i(y,z),i(x,z)))).
P(i(x,i(i(x,y),y))).
P(i(i(x,i(y,z)),i(i(u,y),i(u,i(x,z))))).
P(i(i(x,i(y,i(z,u))),i(z,i(x,i(y,u))))).
P(i(i(x,i(y,z)),i(i(u,y),i(x,i(u,z))))).
P(i(i(i(i(x,y),i(z,y)),u),i(i(z,x),u))).
P(i(i(x,i(y,z)),i(x,i(i(z,u),i(y,u))))).
P(i(i(i(x,i(y,z)),u),i(i(y,i(x,z)),u))).
P(i(i(i(i(x,y),i(x,z)),u),i(i(y,z),u))).
P(i(i(i(i(x,y),y),z),i(x,z))).
P(i(i(x,y),i(x,i(i(y,z),z)))).
P(i(i(i(x,x),y),y)).
P(i(i(x,y),i(i(z,i(y,u)),i(x,i(z,u))))).
P(i(i(x,y),i(i(y,i(z,u)),i(z,i(x,u))))).
P(i(i(x,i(y,z)),i(i(i(x,z),u),i(y,u)))).
```

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

An 8-Step Proof

```
----- Otter 3.3g-work, Jan 2005 -----
The process was started by wos on octopus.mcs.anl.gov,
Wed Feb 25 20:28:59 2009
The command was "otter".  The process ID is 16812.

----> UNIT CONFLICT at   1.12 sec ----> 6394 [binary,6393.1,5.1] $ANS(TARG29).

Length of proof is 8.  Level of proof is 4.
```

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

1 [] -P(i(x,y)) | -P(x) | P(y).
2 [] P(i(i(x,y),i(i(z,x),i(z,y)))).
3 [] P(i(i(x,i(y,z)),i(y,i(x,z)))).
4 [] P(i(x,x)).
5 [] -P(i(i(i(a1,a1),i(a2,i(a3,a4))),i(i(a4,a5),i(a3,i(a2,a5))))) | $ANS(TARG29).
71 [hyper,1,2,2] P(i(i(x,i(y,z)),i(x,i(i(u,y),i(u,z))))).
73 [hyper,1,2,3] P(i(i(x,i(y,i(z,u))),i(x,i(z,i(y,u))))).
79 [hyper,1,71,3] P(i(i(x,i(y,z)),i(i(u,y),i(u,i(x,z))))).
80 [hyper,1,71,2] P(i(i(x,y),i(i(z,i(u,x)),i(z,i(u,y))))).
91 [hyper,1,3,73] P(i(x,i(i(x,i(y,i(z,u))),i(z,i(y,u))))).
164 [hyper,1,79,80] P(i(i(x,i(y,i(z,u))),i(x,i(i(u,v),i(y,i(z,v)))))).
343 [hyper,1,91,4] P(i(i(i(x,x),i(y,i(z,u))),i(z,i(y,u)))).
6393 [hyper,1,164,343] P(i(i(i(x,x),i(y,i(z,u))),i(i(u,v),i(z,i(y,v))))).

The story being told ends—or so it appears—with one additional marvelous development. Ulrich, before Halleck and I began our collaboration, had fifty-seven 5-variable formulas of length 19 that had not yet been classified with regard to being a single axiom for the *BCI* logic. Formula 29 is one of those fifty-seven, which left fifty-six to consider. Ulrich, apparently prompted by the reported success here with formula 29, resumed his study of *BCI*. His effort yielded forty-three additional single axioms, the following.

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

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

There remain (using in part the Ulrich's original numbering) the following thirteen 5-variable length-19 formulas as yet unclassified with regard to being a single axiom. Some cursory experiments led me, at the time of this writing, to believe that none of them has sufficient power to axiomatize, by itself, the *BCI* logic. If any of you succeed in proving, perhaps with a model, that any are insufficient, or prove that any are in fact strong enough, I, Halleck, and Ulrich would each enjoy hearing from you. The thirteen in question are these.

P(i(i(i(u,u),i(v,w)),i(x,i(i(y,i(x,v)),i(y,w))))). % BCI-Candidate 10
P(i(i(u,i(i(v,v),w)),i(i(i(i(x,y),y),u),i(x,w)))). % BCI-Candidate 13
P(i(i(u,v),i(i(i(i(i(w,w),i(v,x)),x),y),i(u,y)))). % BCI-Candidate 18
P(i(i(u,v),i(i(i(i(i(w,w),i(x,u)),v),y),i(x,y)))). % BCI-Candidate 19
P(i(i(u,v),i(i(i(i(w,x),x),i(i(y,y),u)),i(w,v)))). % BCI-Candidate 23
P(i(u,i(i(v,w),i(i(i(x,x),i(y,i(u,v))),i(y,w))))). % BCI-Candidate 42
P(i(i(i(i(i(u,u),i(v,w)),i(i(x,v),w)),y),i(x,y))). % BCI-Candidate 46
P(i(i(i(i(i(u,u),i(v,w)),x),y),i(i(w,x),i(v,y)))). % BCI-Candidate 48
P(i(i(i(i(u,v),i(i(w,u),v)),i(i(x,x),y)),i(w,y))). % BCI-Candidate 51
P(i(i(i(i(u,v),v),i(i(w,w),x)),i(i(x,y),i(u,y)))). % BCI-Candidate 52
P(i(i(i(i(u,v),v),i(i(w,w),x)),i(i(y,u),i(y,x)))). % BCI-Candidate 53
P(i(i(i(i(u,v),w),i(i(x,x),y)),i(i(v,w),i(u,y)))). % BCI-Candidate 54
P(i(i(i(i(u,v),v),w),i(i(w,i(i(x,x),y)),i(u,y)))). % BCI-Candidate 57

After this notbook had been completed, research by me, Halleck, and M. Stickel did in fact prove that eight of the thirteen are single axioms, namely, 10 and 18 by Halleck, 19, 42, and 48 by Stickel, and 23, 51, and 46 by me. That research will probably be reported in yet another notebook. Nevertheless, you might still enjoy studying on your own any or all of the given thirteen.

A story, stirring to at least three researchers, comes to a close—except for an epilogue.

5. Epilogue

The birth of this epilogue rests with an offhand remark made by an associate of Halleck's, then reported to me by Halleck. Numbers mean different things to different people, as is well known. Halleck's associate, learning of the 53-step proof, expressed what might be classed as disappointment, noting that the number of cards in a standard deck is fifty-two. Clearly, that number, fifty-two, was more appealing. Therefore, although I had tried quite hard to find a proof of length strictly less than 53, failing on all paths, I decided to take an approach that sometimes leads to progress.

As you know, in general, where demodulation is used to purge unwanted deductions, the plan is to block steps one at a time of a given proof with the goal of forcing the program to complete a shorter proof. If and when the shorter proof is found, you can continue blocking proof steps, or you can change the focus to the newer and shorter proof and resume using demodulation to purge items one at a time. A variation of this approach, one requiring far more time usually, is that of blocking proof steps two at a time. I seldom try three at a time. I did in fact turn to two-at-a-time blocking.

The focus was the 53-step proof given earlier in this notebook. McCune again came to the rescue. His program otter-loop was used earlier in this notebook to block steps one at a time. I now used another of his programs, otter-loopn, which has you choose how may steps to block at a time, two or more, as it loops through the list of items you wish blocked. And it found a pair of proof steps of the given 53-step proof, the 15th and 16th, that, if blocked at the same time, led to finding a *different* 53-step proof. (I leave to you the discovery of that new 53-step proof if your curiosity is sufficient; it might prove stimulating.) The new 53-step proof contains exactly five formulas not present in the first 53-step proof, each of which relies on three distinct variables. Therefore, the new 53-step proof contains twenty-nine 5-variable formulas, each of which is a single axiom for *BCI*.

I immediately switched my focus to the new 53-step proof, returning to otter-loop to block proof steps one at a time. As progress occurred, I sometimes again switched my focus to the even newer and even shorter proof. Eventually, I had the following 48-step proof.

A Charming 48-Step Proof

```
----- Otter 3.3g-work, Jan 2005 -----
The process was started by wos on octopus.mcs.anl.gov,
Thu Feb 26 15:23:13 2009
The command was "otter".  The process ID is 17284.

-----> EMPTY CLAUSE at   0.08 sec ----> 289 [hyper,2,155,210,159,254,214,226,265,165,193,256,
    207,266,158,195,221,170,217,206,216,215,267,169,133,235,168,140,144,205,208] $ANS(TARGALL29).

Length of proof is 48.  Level of proof is 11.

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

1 [] -P(i(x,y)) | -P(x) | P(y).
2 [] -P(i(i(i(a1,a1),i(a2,i(a3,a4))),i(i(a4,a5),i(a3,i(a2,a5))))) |
 -P(i(i(a1,i(a2,a3)),i(i(i(a4,a4),i(a5,a2)),i(a5,i(a1,a3))))) |
 -P(i(i(i(a1,a1),i(a2,a3)),i(i(a4,i(a3,a5)),i(a2,i(a4,a5))))) |
 -P(i(i(a1,a2),i(i(i(a3,a3),i(i(i(a2,a4),a4),a5)),i(a1,a5)))) |
 -P(i(i(a1,a2),i(i(a2,i(i(a3,a3),i(a4,a5))),i(a4,i(a1,a5))))) |
 -P(i(i(i(i(i(a1,a1),i(a2,a3)),a3),a4),i(i(a4,a5),i(a2,a5)))) |
 -P(i(i(i(i(a1,a2),a2),a3),i(i(i(a4,a4),i(a3,a5)),i(a1,a5)))) |
 -P(i(i(a1,i(a2,a3)),i(i(i(i(i(a4,a4),a1),a3),a5),i(a2,a5)))) |
 -P(i(i(a1,i(a2,a3)),i(i(i(a4,a4),i(a3,a5)),i(a2,i(a1,a5))))) |
 -P(i(i(i(a1,a1),i(i(i(a2,a3),a3),a4)),i(i(a4,a5),i(a2,a5)))) |
 -P(i(i(i(a1,a1),i(a2,a3)),i(a4,i(i(a3,i(a4,a5)),i(a2,a5))))) |
 -P(i(i(i(a1,a1),i(a2,a3)),i(i(i(i(a3,a4),a4),a5),i(a2,a5)))) |
 -P(i(i(i(a1,a1),i(a2,a3)),i(i(a3,i(a4,a5)),i(a4,i(a2,a5))))) |
 -P(i(i(a1,i(i(a2,a2),i(a3,a4))),i(i(a4,a5),i(a3,i(a1,a5))))) |
 -P(i(i(a1,i(i(a2,a2),a3)),i(a4,i(i(a3,i(a4,a5)),i(a1,a5))))) |
 -P(i(i(a1,a2),i(i(i(a3,a3),i(a2,i(a4,a5))),i(a4,i(a1,a5))))) |
 -P(i(i(a1,i(a2,a3)),i(a2,i(i(i(a4,a4),i(a3,a5)),i(a1,a5))))) |
 -P(i(a1,i(i(i(a2,a2),i(a3,a4)),i(i(a4,i(a1,a5)),i(a3,a5))))) |
 -P(i(a1,i(i(a2,i(a1,a3)),i(i(i(a4,a4),i(a3,a5)),i(a2,a5))))) |
 -P(i(i(a1,a2),i(a3,i(i(i(a4,a4),i(a2,i(a3,a5))),i(a1,a5))))) |
 -P(i(i(i(a1,a1),i(a2,a3)),i(i(i(i(a4,a5),a5),a2),i(a4,a3)))) |
 -P(i(a1,i(i(i(a2,a2),i(a3,i(a1,a4))),i(i(a5,a3),i(a5,a4))))) |
 -P(i(i(i(a1,a1),i(a2,i(a3,a4))),i(i(a5,a2),i(a3,i(a5,a4))))) |
 -P(i(i(a1,a2),i(i(i(a3,a3),i(i(i(a4,a1),a2),a5)),i(a4,a5)))) |
```

-P(i(i(a1,i(i(a2,a2),i(a3,a4))),i(i(a5,a1),i(a3,i(a5,a4))))) |
-P(i(i(i(a1,a1),i(a2,a3)),i(i(i(a4,a2),a3),a5),i(a4,a5)))) |
-P(i(i(i(a1,a1),i(a2,a3)),i(i(a4,i(a5,a2)),i(a5,i(a4,a3))))) |
-P(i(i(a1,i(i(a2,a2),a3)),i(i(a3,i(a4,a5)),i(a4,i(a1,a5))))) |
-P(i(i(a1,i(a2,a3)),i(i(i(a4,a4),i(a5,a1)),i(a2,i(a5,a3))))) | $ANS(TARGALL29).
3 [] P(i(i(x,y),i(i(z,x),i(z,y)))).
4 [] P(i(i(x,i(y,z)),i(y,i(x,z)))).
5 [] P(i(x,x)).
62 [hyper,1,3,4] P(i(i(x,i(y,i(z,u))),i(x,i(z,i(y,u))))).
63 [hyper,1,4,3] P(i(i(x,y),i(i(y,z),i(x,z)))).
65 [hyper,1,62,62] P(i(i(x,i(y,i(z,u))),i(z,i(x,i(y,u))))).
70 [hyper,1,3,63] P(i(i(x,i(y,z)),i(x,i(i(z,u),i(y,u))))).
71 [hyper,1,63,4] P(i(i(i(x,i(y,z)),u),i(i(y,i(x,z)),u))).
72 [hyper,1,63,3] P(i(i(i(i(x,y),i(x,z)),u),i(i(y,z),u))).
75 [hyper,1,65,3] P(i(x,i(i(y,z),i(i(x,y),z)))).
80 [hyper,1,70,4] P(i(i(x,i(y,z)),i(i(i(x,z),u),i(y,u)))).
86 [hyper,1,71,3] P(i(i(x,i(y,z)),i(i(u,y),i(u,i(x,z))))).
92 [hyper,1,72,62] P(i(i(x,i(y,z)),i(i(u,x),i(y,i(u,z))))).
97 [hyper,1,75,5] P(i(i(x,y),i(i(i(z,z),x),y))).
98 [hyper,1,72,80] P(i(i(x,y),i(i(i(i(z,x),y),u),i(z,u)))).
115 [hyper,1,72,92] P(i(i(x,y),i(i(z,i(u,x)),i(u,i(z,y))))).
120 [hyper,1,92,3] P(i(i(x,i(y,z)),i(i(u,y),i(x,i(u,z))))).
126 [hyper,1,71,97] P(i(i(x,i(y,z)),i(i(i(u,u),y),i(x,z)))).
129 [hyper,1,63,97] P(i(i(i(i(i(x,x),y),z),u),i(i(y,z),u))).
133 [hyper,1,97,92] P(i(i(i(x,x),i(y,i(z,u))),i(i(v,y),i(z,i(v,u))))).
140 [hyper,1,97,98] P(i(i(i(x,x),i(y,z)),i(i(i(i(u,y),z),v),i(u,v)))).
144 [hyper,1,97,115] P(i(i(i(x,x),i(y,z)),i(i(u,i(v,y)),i(v,i(u,z))))).
155 [hyper,1,126,115] P(i(i(i(x,x),i(y,i(z,u))),i(i(u,v),i(z,i(y,v))))).
158 [hyper,1,126,92] P(i(i(i(x,x),i(y,z)),i(i(z,i(u,v)),i(u,i(y,v))))).
159 [hyper,1,126,86] P(i(i(i(x,x),i(y,z)),i(i(u,i(z,v)),i(y,i(u,v))))).
165 [hyper,1,129,80] P(i(i(x,i(y,z)),i(i(i(i(i(u,u),x),z),v),i(y,v)))).
168 [hyper,1,71,133] P(i(i(x,i(i(y,y),i(z,u))),i(i(v,x),i(z,i(v,u))))).
169 [hyper,1,65,133] P(i(x,i(i(i(y,y),i(z,i(x,u))),i(i(v,z),i(v,u))))).
170 [hyper,1,4,133] P(i(i(x,y),i(i(i(z,z),i(y,i(u,v))),i(u,i(x,v))))).
193 [hyper,1,4,144] P(i(i(x,i(y,z)),i(i(i(u,u),i(z,v)),i(y,i(x,v))))).
195 [hyper,1,71,155] P(i(i(x,i(i(y,y),i(z,u))),i(i(u,v),i(z,i(x,v))))).
205 [hyper,1,71,158] P(i(i(x,i(i(y,y),z)),i(i(z,i(u,v)),i(u,i(x,v))))).
206 [hyper,1,65,158] P(i(x,i(i(i(y,y),i(z,u)),i(i(u,i(x,v)),i(z,v))))).
207 [hyper,1,62,158] P(i(i(i(x,x),i(y,z)),i(u,i(i(z,i(u,v)),i(y,v))))).
208 [hyper,1,4,158] P(i(i(x,i(y,z)),i(i(i(u,u),i(v,x)),i(y,i(v,z))))).
210 [hyper,1,4,159] P(i(i(x,i(y,z)),i(i(i(u,u),i(v,y)),i(v,i(x,z))))).
213 [hyper,1,165,5] P(i(i(i(i(i(x,x),i(y,z)),z),u),i(y,u))).
214 [hyper,1,4,168] P(i(i(x,y),i(i(y,i(i(z,z),i(u,v))),i(u,i(x,v))))).
215 [hyper,1,65,169] P(i(i(x,y),i(z,i(i(i(u,u),i(y,i(z,v))),i(x,v))))).
216 [hyper,1,65,193] P(i(x,i(i(y,i(x,z)),i(i(i(u,u),i(z,v)),i(y,v))))).
217 [hyper,1,62,193] P(i(i(x,i(y,z)),i(y,i(i(i(u,u),i(z,v)),i(x,v))))).
221 [hyper,1,62,205] P(i(i(x,i(i(y,y),z)),i(u,i(i(z,i(u,v)),i(x,v))))).
226 [hyper,1,70,213] P(i(i(i(i(i(x,x),i(y,z)),z),u),i(i(u,v),i(y,v)))).
235 [hyper,1,217,75] P(i(i(x,y),i(i(i(z,z),i(i(i(u,x),y),v)),i(u,v)))).
251 [hyper,1,235,5] P(i(i(i(x,x),i(i(i(y,z),z),u)),i(y,u))).
252 [hyper,1,251,251] P(i(x,i(i(i(i(x,y),y),z),z))).
254 [hyper,1,120,251] P(i(i(x,y),i(i(i(z,z),i(i(i(y,u),u),v)),i(x,v)))).
256 [hyper,1,70,251] P(i(i(i(x,x),i(i(i(y,z),z),u)),i(i(u,v),i(y,v)))).

```
265 [hyper,1,217,252] P(i(i(i(i(x,y),y),z),i(i(i(u,u),i(z,v)),i(x,v)))).
266 [hyper,1,208,252] P(i(i(i(x,x),i(y,z)),i(i(i(i(z,u),u),v),i(y,v)))).
267 [hyper,1,193,252] P(i(i(i(x,x),i(y,z)),i(i(i(i(u,v),v),y),i(u,z)))).
```

The following input file yields the just-given 48-step proof; it may be of use, especially if you attempt to improve on the proof by finding a still shorter one.

Input 6

```
set(hyper_res).
assign(max_weight,19).
assign(max_seconds,1).
% assign(change_limit_after,800).
% assign(new_max_weight,22).
assign(max_proofs,14).
clear(print_kept).
% set(process_input).
set(ancestor_subsume).
set(back_sub).
% clear(for_sub).
clear(print_back_sub).
clear(print_kept).
clear(print_new_demod).
clear(print_back_demod).
clear(print_back_sub).
% clear(print_given).
assign(max_distinct_vars,8).
assign(pick_given_ratio,10).
assign(max_mem,750000).
assign(report,5400).
set(order_history).
set(input_sos_first).
%  set(sos_queue).
% set(sos_stack).
assign(bsub_hint_wt,1).
set(keep_hint_subsumers).
assign(heat,0).
% assign(dynamic_heat_weight,2).

weight_list(pick_and_purge).
%  weight(P(i(i(x,y),i(z,i(i(z,x),y)))),1000).
weight(junk,1000).
end_of_list.

list(usable).
-P(i(x,y)) | -P(x) | P(y).
-P(i(i(i(a1,a1),i(a2,i(a3,a4))),i(i(a4,a5),i(a3,i(a2,a5))))) | -P(i(i(a1,i(a2,a3)),i(i(i(a4,a4),
    i(a5,a2)),i(a5,i(a1,a3))))) | -P(i(i(i(a1,a1),i(a2,a3)),i(i(a4,i(a3,a5)),i(a2,i(a4,a5))))) |
    -P(i(i(a1,a2),i(i(i(a3,a3),i(i(i(a2,a4),a4),a5)),i(a1,a5)))) | -P(i(i(a1,a2),i(i(a2,i(i(a3,a3),
    i(a4,a5))),i(a4,i(a1,a5))))) | -P(i(i(i(i(i(a1,a1),i(a2,a3)),a3),a4),i(i(a4,a5),i(a2,a5)))) |
    -P(i(i(i(i(a1,a2),a2),a3),i(i(i(a4,a4),i(a3,a5)),i(a1,a5)))) | -P(i(i(a1,i(a2,a3)),
    i(i(i(i(i(a4,a4),a1),a3),a5),i(a2,a5)))) | -P(i(i(a1,i(a2,a3)),i(i(i(a4,a4),i(a3,a5)),
    i(a2,i(a1,a5))))) | -P(i(i(i(a1,a1),i(i(i(a2,a3),a3),a4)),i(i(a4,a5),i(a2,a5)))) |
```

```
    -P(i(i(i(a1,a1),i(a2,a3)),i(a4,i(i(a3,i(a4,a5)),i(a2,a5))))) | -P(i(i(i(a1,a1),i(a2,a3)),
    i(i(i(i(a3,a4),a4),a5),i(a2,a5)))) | -P(i(i(i(a1,a1),i(a2,a3)),i(i(a3,i(a4,a5)),i(a4,i(a2,a5))))) |
    -P(i(i(a1,i(i(a2,a2),i(a3,a4))),i(i(a4,a5),i(a3,i(a1,a5))))) | -P(i(i(a1,i(i(a2,a2),a3)),
    i(a4,i(i(a3,i(a4,a5)),i(a1,a5))))) | -P(i(i(a1,a2),i(i(i(a3,a3),i(a2,i(a4,a5))),i(a4,i(a1,a5))))) |
    -P(i(i(a1,i(a2,a3)),i(a2,i(i(i(a4,a4),i(a3,a5)),i(a1,a5))))) | -P(i(a1,i(i(i(a2,a2),i(a3,a4)),
    i(i(a4,i(a1,a5)),i(a3,a5))))) | -P(i(a1,i(i(a2,i(a1,a3)),i(i(i(a4,a4),i(a3,a5)),i(a2,a5))))) |
    -P(i(i(a1,a2),i(a3,i(i(i(a4,a4),i(a2,i(a3,a5))),i(a1,a5))))) | -P(i(i(i(a1,a1),i(a2,a3)),
    i(i(i(i(a4,a5),a5),a2),i(a4,a3)))) | -P(i(a1,i(i(i(a2,a2),i(a3,i(a1,a4))),i(i(a5,a3),i(a5,a4))))) |
    -P(i(i(i(a1,a1),i(a2,i(a3,a4))),i(i(a5,a2),i(a3,i(a5,a4))))) | -P(i(i(a1,a2),i(i(i(a3,a3),
    i(i(i(a4,a1),a2),a5)),i(a4,a5)))) | -P(i(i(a1,i(i(a2,a2),i(a3,a4))),i(i(a5,a1),i(a3,i(a5,a4))))) |
    -P(i(i(i(a1,a1),i(a2,a3)),i(i(i(i(a4,a2),a3),a5),i(a4,a5)))) | -P(i(i(i(a1,a1),i(a2,a3)),
    i(i(a4,i(a5,a2)),i(a5,i(a4,a3))))) | -P(i(i(a1,i(i(a2,a2),a3)),i(i(a3,i(a4,a5)),i(a4,i(a1,a5))))) |
    -P(i(i(a1,i(a2,a3)),i(i(i(a4,a4),i(a5,a1)),i(a2,i(a5,a3))))) | $ANS(TARGALL29).
%  -P(i(i(a1,i(b,a2)),i(b,i(a1,a2)))) | -P(i(a1,a1)) | -P(i(i(a1,b),i(i(a2,a1),i(a2,b)))) |
    $ANS(all).  % BCI
end_of_list.

list(sos).
P(i(i(x,y),i(i(z,x),i(z,y)))).  % B
P(i(i(x,i(y,z)),i(y,i(x,z)))).  % C
P(i(x,x)).  % I
end_of_list.

list(passive).
%  Following two Meredith single axioms for BCI.
-P(i(i(a,i(b,c)),i(i(i(d,d),i(e,b)),i(e,i(a,c))))) | $ANS(MER).
-P(i(i(i(a,a),i(b,c)),i(i(d,i(c,e)),i(b,i(d,e))))) | $ANS(MER).
%  Following is neg of the newest single found by Halleck and Wos, the 29th.
-P(i(i(i(a1,a1),i(a2,i(a3,a4))),i(i(a4,a5),i(a3,i(a2,a5))))) | $ANS(TARG29).
    %  a 29th not among the 28 known singles for BCI.
end_of_list.

%  Following purges unwanted formulas.
list(demodulators).
(P(i(i(x,i(y,z)),i(x,i(i(u,y),i(u,z))))) = junk).
% (P(i(i(i(i(x,y),y),z),i(x,z))) = junk).
%  (P(i(i(x,y),i(x,i(i(y,z),z)))) = junk).
%  (P(i(i(i(x,x),y),y)) = junk).
%  (P(i(i(x,y),i(i(i(i(y,z),z),u),i(x,u)))) = junk).
%  (P(i(i(x,y),i(i(i(z,z),i(y,u)),i(x,u)))) = junk).
%  older demods
%  (P(i(i(i(i(x,y),y),z),i(x,z))) = junk).
%  (P(i(i(x,y),i(i(z,i(y,u)),i(x,i(z,u))))) = junk).
%  (P(i(i(x,i(y,z)),i(x,i(i(i(u,u),i(z,v)),i(y,v))))) = junk).
%  (P(i(i(x,y),i(i(y,i(z,u)),i(z,i(x,u))))) = 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.

list(hints).
```

% Following 48/11 purport to prove the join of the 28 singles for BCI plus the 29th, temp.bci.halleck.out1u.
P(i(i(x,i(y,i(z,u))),i(x,i(z,i(y,u))))).
P(i(i(x,y),i(i(y,z),i(x,z)))).
P(i(i(x,i(y,i(z,u))),i(z,i(x,i(y,u))))).
P(i(i(x,i(y,z)),i(x,i(i(z,u),i(y,u))))).
P(i(i(i(x,i(y,z)),u),i(i(y,i(x,z)),u))).
P(i(i(i(i(x,y),i(x,z)),u),i(i(y,z),u))).
P(i(x,i(i(y,z),i(i(x,y),z)))).
P(i(i(x,i(y,z)),i(i(i(x,z),u),i(y,u)))).
P(i(i(x,i(y,z)),i(i(u,y),i(u,i(x,z))))).
P(i(i(x,i(y,z)),i(i(u,x),i(y,i(u,z))))).
P(i(i(x,y),i(i(i(z,z),x),y))).
P(i(i(x,y),i(i(i(i(z,x),y),u),i(z,u)))).
P(i(i(x,y),i(i(z,i(u,x)),i(u,i(z,y))))).
P(i(i(x,i(y,z)),i(i(u,y),i(x,i(u,z))))).
P(i(i(x,i(y,z)),i(i(i(u,u),y),i(x,z)))).
P(i(i(i(i(i(x,x),y),z),u),i(i(y,z),u))).
P(i(i(i(x,x),i(y,i(z,u))),i(i(v,y),i(z,i(v,u))))).
P(i(i(i(x,x),i(y,z)),i(i(i(i(u,y),z),v),i(u,v)))).
P(i(i(i(x,x),i(y,z)),i(i(u,i(v,y)),i(v,i(u,z))))).
P(i(i(i(x,x),i(y,i(z,u))),i(i(u,v),i(z,i(y,v))))).
P(i(i(i(x,x),i(y,z)),i(i(z,i(u,v)),i(u,i(y,v))))).
P(i(i(i(x,x),i(y,z)),i(i(u,i(z,v)),i(y,i(u,v))))).
P(i(i(x,i(y,z)),i(i(i(i(i(u,u),x),z),v),i(y,v)))).
P(i(i(x,i(i(y,y),i(z,u))),i(i(v,x),i(z,i(v,u))))).
P(i(x,i(i(i(y,y),i(z,i(x,u))),i(i(v,z),i(v,u))))).
P(i(i(x,y),i(i(i(z,z),i(y,i(u,v))),i(u,i(x,v))))).
P(i(i(x,i(y,z)),i(i(i(u,u),i(z,v)),i(y,i(x,v))))).
P(i(i(x,i(i(y,y),i(z,u))),i(i(u,v),i(z,i(x,v))))).
P(i(i(x,i(i(y,y),z)),i(i(z,i(u,v)),i(u,i(x,v))))).
P(i(x,i(i(i(y,y),i(z,u)),i(i(u,i(x,v)),i(z,v))))).
P(i(i(i(x,x),i(y,z)),i(u,i(i(z,i(u,v)),i(y,v))))).
P(i(i(x,i(y,z)),i(i(i(u,u),i(v,x)),i(y,i(v,z))))).
P(i(i(x,i(y,z)),i(i(i(u,u),i(v,y)),i(v,i(x,z))))).
P(i(i(i(i(i(x,x),i(y,z)),z),u),i(y,u))).
P(i(i(x,y),i(i(y,i(i(z,z),i(u,v))),i(u,i(x,v))))).
P(i(i(x,y),i(z,i(i(i(u,u),i(y,i(z,v))),i(x,v))))).
P(i(x,i(i(y,i(x,z)),i(i(i(u,u),i(z,v)),i(y,v))))).
P(i(i(x,i(y,z)),i(y,i(i(i(u,u),i(z,v)),i(x,v))))).
P(i(i(x,i(i(y,y),z)),i(u,i(i(z,i(u,v)),i(x,v))))).
P(i(i(i(i(i(x,x),i(y,z)),z),u),i(i(u,v),i(y,v)))).
P(i(i(x,y),i(i(i(z,z),i(i(i(u,x),y),v)),i(u,v)))).
P(i(i(i(x,x),i(i(i(y,z),z),u)),i(y,u))).
P(i(x,i(i(i(i(x,y),y),z),z))).
P(i(i(x,y),i(i(i(z,z),i(i(i(y,u),u),v)),i(x,v)))).
P(i(i(i(x,x),i(i(i(y,z),z),u)),i(i(u,v),i(y,v)))).
P(i(i(i(i(x,y),y),z),i(i(i(u,u),i(z,v)),i(x,v)))).
P(i(i(i(x,x),i(y,z)),i(i(i(i(z,u),u),v),i(y,v)))).
P(i(i(i(x,x),i(y,z)),i(i(i(i(u,v),v),y),i(u,z)))).
end_of_list.

Of interest, at least to me, all of the forty-eight steps of the 48-step proof are among those of the new 53-step proof. Two of the forty-eight are not among those of the first 53-step proof, each relying on three distinct variables. I thus have a proof containing twenty-nine single axioms and no other 5-variable formula. (As an aside, you might find it amusing to note that 48 is the number of cards in a pinochle deck of cards, and, as observed, the fact that the standard deck of cards contains 52 played an odd role in the story central to this notebook.) Can you find a shorter proof?

You thus see what the general approach is for proof shortening, one of iteration, where demodulation is used to block the use of formulas deemed to be unwanted. Specifically, you focus on a proof in hand, systematically block each of its deduced steps (with demodulation), and, when blocking a step results in the completion of a shorter proof, block another step of the proof in hand to see what happens. At some point, you switch the focus to a newer and shorter proof, and resume blocking its steps. You can, as I did at one point, block proof steps two at a time. I seldom block steps three at a time, mostly avoiding this action because of impatience when faced with the large number of cases to consider. But, from an algorithmic viewpoint, from what I have determined from years of experimentation, proof shortening is an art, not a science. The number of possible paths to pursue is in general daunting, if you choose to be extremely thorough. After all, the values assigned to various parameters, when using OTTER, affect the outcome of experimentation, values for the pick_given_ratio, for max_weight, for max_distinct_vars, for example. Perhaps disappointing, but not surprising, in most cases a proof establishing that the shortest possible proof is in hand is hard, or impossible, to come by. If you have a choice to be made between two possible steps to block, each of which yields some progress, but the same amount, I usually choose that which occurs later in the proof in hand. My reason, intuitively, is that the later the step occurs, the more time the program has to (so-to-speak) replace it. If the suggested step to block is the first or second in the proof, you have a rather strong indication that very little additional progress will occur along the given path. I do believe that my colleague and friend Ross Overbeek is correct in his view that much more automation is possible, perhaps based on a study of the various notebooks I offer on my website, automated reasoning.net.

Halleck had the following intriguing notion. From the viewpoint of relying on OTTER, place all of the deduced steps of the 48-step proof in list(sos), together with the 3-axiom system for the *BCI* logic, and apply level saturation (breadth-first) by using set(sos_queue). The goal is to see how many formulas among the known 72 single axioms are deduced at which level. You might recall that an input item has level 0 and that a deduced item has level *n* if at least one of its parents has level *n*-1. At level 1, twenty-six single axioms (not among the twenty-nine found in the 48-step proof) are deduced. In other words, you have in hand a 74-step proof, fifty-five of whose (deduced) steps are known single axioms. For Halleck and for me, such a proof is astounding in its density of powerful formulas; I intend to explain, thanks to Ulrich, how this phenomenon can occur. By the time level 3 is fully examined, all seventy-two known single axioms are present.

Halleck's enthusiasm in the context of the 48-step proof that deduces the join of twenty-nine single axioms from *B*, *C*, and *I* led him, at least in part, to a new project. Specifically, he (with his program shotgun) found a proof of length 123 that deduces from the featured 3-axiom system the join of all seventy-two known single axioms. He sent me this impressively short, 123-step proof with the goal (of the new project) that of finding a shorter proof. As usual, I paid no attention to the history of each deduced step and, instead, simply focused on the deduced formulas. The use of ancestor subsumption alone yielded nothing of interest. I, therefore, turned (as expected) to the use of demodulation to block formulas of the Halleck proof.

Eventually, if memory serves, I had a proof of length less than 100, but could get no further. I turned to a form of cramming. In particular, I had OTTER produce proofs of each of the last few steps of a proof of length, I believe, 99. Among those proofs, I selected a 17-step proof of the last 5-variable formula, a single axiom. I added to the three basic axioms the seventeen steps of the chosen proof, placing them in list(sos), and, surprisingly, did not use level saturation. OTTER found an 83-step proof, relying on the given twenty formulas, that proved the join of the 72 known single axioms. With that 100-step proof as the focus of attention, demodulation blocking eventually, I think, led to the finding of a 94-step proof, which was most pleasing to Halleck.

Halleck during this series of experiments had sent to Ulrich a 98-step proof I had found and, although I did not know at the time, Ulrich used his program and his methodology to discover a 92-step proof.

Ulrich does, in contrast to me, examine the history of deduced conclusions and applies various approaches he has devised. I am indeed impressed with Ulrich's success.

I took his 92-step proof, just the deduced formulas, and used Veroff's hints strategy with demodulation blocking and eventually had the following 90-step proof. The proof deduces from the 3-axiom system featured throughout the join of the known single axioms, seventy-two of them, and has the property that it contains two additional 5-variable formulas.

A Delightful 90-Step Proof

----- Otter 3.3g-work, Jan 2005 -----
The process was started by wos on elephant.mcs.anl.gov,
Wed Mar 11 07:59:10 2009
The command was "otter". The process ID is 24304.

-----> EMPTY CLAUSE at 0.73 sec ----> 200 [hyper,2,139,127,193,161,124,137,155,191,125,
133,188,123,181,150,142,184,152,192,163,126,162,128,157,140,189,195,132,134,158,196,
176,172,122,182,141,130,138,156,129,151,154,159,169,136,198,175,131,153,143,168,146,
171,194,187,148,197,178,166,167,170,177,180,165,179,164,185,199,186,173,174,183,190]
$ANS(TARG2).

Length of proof is 90. Level of proof is 11.

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

1 [] -P(i(x,y)) | -P(x) | P(y).
2 [] -P(i(i(p,i(q,r)),i(i(i(s,s),i(t,q)),i(t,i(p,r))))) |
-P(i(i(i(p,p),i(q,r)),i(i(s,i(r,t)),i(q,i(s,t))))) |
-P(i(i(p,q),i(i(i(r,r),i(i(q,s),s),t)),i(p,t)))) |
-P(i(i(p,q),i(i(q,i(i(r,r),i(s,t))),i(s,i(p,t))))) |
-P(i(i(i(i(i(p,p),i(q,r)),r),s),i(i(s,t),i(q,t)))) |
-P(i(i(i(i(p,q),q),r),i(i(i(s,s),i(r,t)),i(p,t)))) |
-P(i(i(p,i(q,r)),i(i(i(i(s,s),p),r),t),i(q,t)))) |
-P(i(i(p,i(q,r)),i(i(i(s,s),i(r,t)),i(q,i(p,t))))) |
-P(i(i(i(p,p),i(i(i(q,r),r),s)),i(i(s,t),i(q,t)))) |
-P(i(i(i(p,p),i(q,r)),i(s,i(i(r,i(s,t)),i(q,t))))) |
-P(i(i(i(p,p),i(q,r)),i(i(i(i(r,s),s),t),i(q,t)))) |
-P(i(i(i(p,p),i(q,r)),i(i(r,i(s,t)),i(s,i(q,t))))) |
-P(i(i(p,i(i(q,q),i(r,s))),i(i(s,t),i(r,i(p,t))))) |
-P(i(i(p,i(i(q,q),r)),i(s,i(i(r,i(s,t)),i(p,t))))) |
-P(i(i(p,q),i(i(i(r,r),i(q,i(s,t))),i(s,i(p,t))))) |
-P(i(i(p,i(q,r)),i(q,i(i(i(s,s),i(r,t)),i(p,t))))) |
-P(i(p,i(i(i(q,q),i(r,s)),i(i(s,i(p,t)),i(r,t))))) |
-P(i(p,i(i(q,i(p,r)),i(i(i(s,s),i(r,t)),i(q,t))))) |
-P(i(i(p,q),i(r,i(i(i(s,s),i(q,i(r,t))),i(p,t))))) |
-P(i(i(i(p,p),i(q,r)),i(i(i(i(s,t),t),q),i(s,r)))) |
-P(i(p,i(i(i(q,q),i(r,i(p,s))),i(i(t,r),i(t,s))))) |
-P(i(i(i(p,p),i(q,i(r,s))),i(i(t,q),i(r,i(t,s))))) |
-P(i(i(p,q),i(i(i(r,r),i(i(i(s,p),q),t)),i(s,t)))) |
-P(i(i(p,i(i(q,q),i(r,s))),i(i(t,p),i(r,i(t,s))))) |
-P(i(i(i(p,p),i(q,r)),i(i(i(i(s,q),r),t),i(s,t)))) |
-P(i(i(i(p,p),i(q,r)),i(i(s,i(t,q)),i(t,i(s,r))))) |
-P(i(i(p,i(i(q,q),r)),i(i(r,i(s,t)),i(s,i(p,t))))) |

-P(i(i(p,i(q,r)),i(i(i(s,s),i(t,p)),i(q,i(t,r))))) |
-P(i(i(i(p,p),i(i(i(q,r),i(i(s,q),r)),t)),i(s,t))) |
-P(i(i(i(p,p),i(i(i(q,r),r),s)),i(i(t,q),i(t,s)))) |
-P(i(i(i(p,p),i(i(i(q,r),s),t)),i(i(r,s),i(q,t)))) |
-P(i(i(i(p,p),i(q,i(r,s))),i(i(s,t),i(r,i(q,t))))) |
-P(i(i(i(p,p),i(q,i(r,s))),i(i(t,r),i(t,i(q,s))))) |
-P(i(i(i(p,p),i(q,i(r,s))),i(r,i(i(s,t),i(q,t))))) |
-P(i(i(i(p,p),i(q,i(r,s))),i(r,i(i(t,q),i(t,s))))) |
-P(i(i(i(p,p),i(q,i(r,s))),i(t,i(i(t,r),i(q,s))))) |
-P(i(i(i(p,p),i(q,r)),i(q,i(i(s,i(r,t)),i(s,t))))) |
-P(i(i(i(p,p),q),i(i(i(q,i(r,s)),s),t),i(r,t)))) |
-P(i(i(p,i(i(q,q),i(r,s))),i(i(t,r),i(t,i(p,s))))) |
-P(i(i(p,i(q,r)),i(i(s,i(i(t,t),p)),i(q,i(s,r))))) |
-P(i(i(p,i(q,r)),i(q,i(i(i(s,s),i(t,p)),i(t,r))))) |
-P(i(i(p,i(q,r)),i(s,i(i(i(t,t),i(s,q)),i(p,r))))) |
-P(i(i(p,i(q,r)),i(q,i(i(s,i(i(t,t),p)),i(s,r))))) |
-P(i(i(p,q),i(i(i(i(r,r),i(s,t)),t),p),i(s,q)))) |
-P(i(i(p,q),i(i(i(i(q,r),r),i(i(s,s),t)),i(p,t)))) |
-P(i(i(p,q),i(i(i(i(r,p),q),i(i(s,s),t)),i(r,t)))) |
-P(i(i(p,q),i(i(i(r,r),i(s,i(q,t))),i(p,i(s,t))))) |
-P(i(i(p,q),i(i(i(r,r),i(s,i(t,p))),i(t,i(s,q))))) |
-P(i(i(p,q),i(i(r,i(i(s,s),i(q,t))),i(p,i(r,t))))) |
-P(i(i(p,q),i(i(r,i(i(s,s),i(t,p))),i(t,i(r,q))))) |
-P(i(i(p,q),i(p,i(i(i(r,r),i(s,i(q,t))),i(s,t))))) |
-P(i(i(p,q),i(r,i(i(i(s,s),i(t,i(r,p))),i(t,q))))) |
-P(i(p,i(i(q,i(i(r,r),i(p,s))),i(i(s,t),i(q,t))))) |
-P(i(p,i(i(i(q,q),i(r,i(p,s))),i(i(s,t),i(r,t))))) |
-P(i(p,i(i(i(q,q),i(r,i(s,t))),i(i(p,s),i(r,t))))) |
-P(i(p,i(i(i(q,q),i(r,s)),i(i(t,i(p,r)),i(t,s))))) |
-P(i(p,i(i(q,i(i(r,r),i(p,s))),i(i(t,q),i(t,s))))) |
-P(i(p,i(i(q,i(i(r,r),i(s,t))),i(i(p,s),i(q,t))))) |
-P(i(p,i(i(q,i(i(r,r),s)),i(i(s,i(p,t)),i(q,t))))) |
-P(i(p,i(i(q,i(p,r)),i(i(i(s,s),i(t,q)),i(t,r))))) |
-P(i(p,i(i(q,i(r,s)),i(i(i(t,t),i(p,r)),i(q,s))))) |
-P(i(p,i(i(q,i(p,r)),i(i(s,i(i(t,t),q)),i(s,r))))) |
-P(i(p,i(i(p,q),i(i(i(r,r),i(s,i(q,t))),i(s,t))))) |
-P(i(p,i(i(q,r),i(i(i(s,s),i(r,i(p,t))),i(q,t))))) |
-P(i(p,i(i(p,q),i(i(r,i(i(s,s),i(q,t))),i(r,t))))) |
-P(i(p,i(i(q,r),i(i(r,i(i(s,s),i(p,t))),i(q,t))))) |
-P(i(p,i(i(q,r),i(i(s,i(i(t,t),i(p,q))),i(s,r))))) |
-P(i(i(i(i(i(p,p),i(q,r)),r),s),i(i(t,q),i(t,s)))) |
-P(i(i(i(i(i(p,p),q),r),s),i(i(q,i(t,r)),i(t,s)))) |
-P(i(i(i(i(p,i(q,r)),r),s),i(i(i(t,t),p),i(q,s)))) |
-P(i(i(i(i(p,q),q),r),i(i(i(s,s),i(t,p)),i(t,r)))) |
-P(i(i(i(i(p,q),r),s),i(i(i(t,t),i(q,r)),i(p,s)))) | $ANS(TARG2).
3 [] P(i(i(x,y),i(i(z,x),i(z,y)))).
4 [] P(i(i(x,i(y,z)),i(y,i(x,z)))).
5 [] P(i(x,x)).
106 [hyper,1,3,4] P(i(i(x,i(y,i(z,u))),i(x,i(z,i(y,u))))).
107 [hyper,1,4,3] P(i(i(x,y),i(i(y,z),i(x,z)))).
108 [hyper,1,4,5] P(i(x,i(i(x,y),y))).
109 [hyper,1,107,107] P(i(i(i(i(x,y),i(z,y)),u),i(i(z,x),u))).
110 [hyper,1,3,107] P(i(i(x,i(y,z)),i(x,i(i(z,u),i(y,u))))).

111 [hyper,1,107,4] P(i(i(i(x,i(y,z)),u),i(i(y,i(x,z)),u))).
112 [hyper,1,107,3] P(i(i(i(i(x,y),i(x,z)),u),i(i(y,z),u))).
113 [hyper,1,108,5] P(i(i(i(x,x),y),y)).
114 [hyper,1,109,106] P(i(i(x,y),i(i(y,i(z,u)),i(z,i(x,u))))).
115 [hyper,1,109,110] P(i(i(x,y),i(i(y,z),i(i(z,u),i(x,u))))).
116 [hyper,1,3,111] P(i(i(x,i(i(y,i(z,u)),v)),i(x,i(i(z,i(y,u)),v)))).
117 [hyper,1,111,3] P(i(i(x,i(y,z)),i(i(u,y),i(u,i(x,z))))).
118 [hyper,1,107,113] P(i(i(x,y),i(i(i(z,z),x),y))).
119 [hyper,1,4,113] P(i(x,i(i(i(y,y),i(x,z)),z))).
120 [hyper,1,115,108] P(i(i(i(i(x,y),y),z),i(i(z,u),i(x,u)))).
121 [hyper,1,111,118] P(i(i(x,i(y,z)),i(i(i(u,u),y),i(x,z)))).
122 [hyper,1,118,117] P(i(i(i(x,x),i(y,i(z,u))),i(i(v,z),i(v,i(y,u))))).
123 [hyper,1,118,114] P(i(i(i(x,x),i(y,z)),i(i(z,i(u,v)),i(u,i(y,v))))).
124 [hyper,1,115,119] P(i(i(i(i(i(x,x),i(y,z)),z),u),i(i(u,v),i(y,v)))).
125 [hyper,1,118,120] P(i(i(i(x,x),i(i(i(y,z),z),u)),i(i(u,v),i(y,v)))).
126 [hyper,1,121,120] P(i(i(i(x,x),i(y,z)),i(i(i(i(u,v),v),y),i(u,z)))).
127 [hyper,1,121,117] P(i(i(i(x,x),i(y,z)),i(i(u,i(z,v)),i(y,i(u,v))))).
128 [hyper,1,121,114] P(i(i(i(x,x),i(y,i(z,u))),i(i(v,y),i(z,i(v,u))))).
129 [hyper,1,111,122] P(i(i(x,i(i(y,y),i(z,u))),i(i(v,z),i(v,i(x,u))))).
130 [hyper,1,106,122] P(i(i(i(x,x),i(y,i(z,u))),i(v,i(i(v,z),i(y,u))))).
131 [hyper,1,4,122] P(i(i(x,y),i(i(i(z,z),i(u,i(y,v))),i(x,i(u,v))))).
132 [hyper,1,111,123] P(i(i(x,i(i(y,y),z)),i(i(z,i(u,v)),i(u,i(x,v))))).
133 [hyper,1,106,123] P(i(i(i(x,x),i(y,z)),i(u,i(i(z,i(u,v)),i(y,v))))).
134 [hyper,1,4,123] P(i(i(x,i(y,z)),i(i(i(u,u),i(v,x)),i(y,i(v,z))))).
135 [hyper,1,109,124] P(i(i(x,i(i(y,y),i(z,u))),i(i(i(x,u),v),i(z,v)))).
136 [hyper,1,4,124] P(i(i(x,y),i(i(i(i(i(z,z),i(u,v)),v),x),i(u,y)))).
137 [hyper,1,4,126] P(i(i(i(i(x,y),y),z),i(i(i(u,u),i(z,v)),i(x,v)))).
138 [hyper,1,106,127] P(i(i(i(x,x),i(y,z)),i(y,i(i(u,i(z,v)),i(u,v))))).
139 [hyper,1,4,127] P(i(i(x,i(y,z)),i(i(i(u,u),i(v,y)),i(v,i(x,z))))).
140 [hyper,1,111,128] P(i(i(x,i(i(y,y),i(z,u))),i(i(v,x),i(z,i(v,u))))).
141 [hyper,1,106,128] P(i(i(i(x,x),i(y,i(z,u))),i(z,i(i(v,y),i(v,u))))).
142 [hyper,1,4,128] P(i(i(x,y),i(i(i(z,z),i(y,i(u,v))),i(u,i(x,v))))).
143 [hyper,1,4,129] P(i(i(x,y),i(i(z,i(i(u,u),i(y,v))),i(x,i(z,v))))).
146 [hyper,1,129,122] P(i(i(x,y),i(x,i(i(i(z,z),i(u,i(y,v))),i(u,v))))).
147 [hyper,1,113,130] P(i(x,i(i(x,y),i(i(y,z),z)))).
148 [hyper,1,4,130] P(i(x,i(i(i(y,y),i(z,i(u,v))),i(i(x,u),i(z,v))))).
150 [hyper,1,106,132] P(i(i(x,i(i(y,y),z)),i(u,i(i(z,i(u,v)),i(x,v))))).
151 [hyper,1,4,132] P(i(i(x,i(y,z)),i(i(u,i(i(v,v),x)),i(y,i(u,z))))).
152 [hyper,1,4,133] P(i(x,i(i(i(y,y),i(z,u)),i(i(u,i(x,v)),i(z,v))))).
153 [hyper,1,112,134] P(i(i(x,y),i(i(i(z,z),i(u,i(v,x))),i(v,i(u,y))))).
154 [hyper,1,106,134] P(i(i(x,i(y,z)),i(y,i(i(i(u,u),i(v,x)),i(v,z))))).
155 [hyper,1,112,135] P(i(i(x,i(y,z)),i(i(i(i(i(u,u),x),z),v),i(y,v)))).
156 [hyper,1,109,135] P(i(i(i(x,x),y),i(i(i(i(y,i(z,u)),u),v),i(z,v)))).
157 [hyper,1,112,137] P(i(i(x,y),i(i(i(z,z),i(i(i(u,x),y),v)),i(u,v)))).
158 [hyper,1,137,107] P(i(i(i(x,x),i(i(i(y,z),i(i(u,y),z)),v)),i(u,v))).
159 [hyper,1,106,139] P(i(i(x,i(y,z)),i(u,i(i(i(v,v),i(u,y)),i(x,z))))).
161 [hyper,1,4,140] P(i(i(x,y),i(i(y,i(i(z,z),i(u,v))),i(u,i(x,v))))).
162 [hyper,1,4,141] P(i(x,i(i(i(y,y),i(z,i(x,u))),i(i(v,z),i(v,u))))).
163 [hyper,1,106,142] P(i(i(x,y),i(z,i(i(i(u,u),i(y,i(z,v))),i(x,v))))).
164 [hyper,1,130,143] P(i(x,i(i(x,y),i(i(z,i(i(u,u),i(y,v))),i(z,v))))).
165 [hyper,1,4,146] P(i(x,i(i(x,y),i(i(i(z,z),i(u,i(y,v))),i(u,v))))).
166 [hyper,1,116,148] P(i(x,i(i(y,i(i(z,z),i(u,v))),i(i(x,u),i(y,v))))).
167 [hyper,1,4,150] P(i(x,i(i(y,i(i(z,z),u)),i(i(u,i(x,v)),i(y,v))))).

168 [hyper,1,112,151] P(i(i(x,y),i(i(z,i(i(u,u),i(v,x))),i(v,i(z,y))))).
169 [hyper,1,106,151] P(i(i(x,i(y,z)),i(y,i(i(u,i(i(v,v),x)),i(u,z))))).
170 [hyper,1,106,152] P(i(x,i(i(y,i(x,z)),i(i(i(u,u),i(v,y)),i(v,z))))).
171 [hyper,1,106,153] P(i(i(x,y),i(z,i(i(i(u,u),i(v,i(z,x))),i(v,y))))).
172 [hyper,1,4,153] P(i(i(i(x,x),i(y,i(z,u))),i(i(u,v),i(z,i(y,v))))).
173 [hyper,1,4,155] P(i(i(i(i(i(x,x),y),z),u),i(i(y,i(v,z)),i(v,u)))).
174 [hyper,1,4,156] P(i(i(i(i(x,i(y,z)),z),u),i(i(i(v,v),x),i(y,u)))).
175 [hyper,1,116,157] P(i(i(x,y),i(i(i(i(z,x),y),i(i(u,u),v)),i(z,v)))).
176 [hyper,1,4,157] P(i(i(i(x,x),i(i(i(y,z),u),v)),i(i(z,u),i(y,v)))).
177 [hyper,1,4,159] P(i(x,i(i(y,i(z,u)),i(i(i(v,v),i(x,z)),i(y,u))))).
178 [hyper,1,116,162] P(i(x,i(i(y,i(i(z,z),i(x,u))),i(i(v,y),i(v,u))))).
179 [hyper,1,106,162] P(i(x,i(i(y,z),i(i(i(u,u),i(z,i(x,v))),i(y,v))))).
180 [hyper,1,106,167] P(i(x,i(i(y,i(x,z)),i(i(u,i(i(v,v),y)),i(u,z))))).
181 [hyper,1,4,168] P(i(i(x,i(i(y,y),i(z,u))),i(i(u,v),i(z,i(x,v))))).
182 [hyper,1,106,172] P(i(i(i(x,x),i(y,i(z,u))),i(z,i(i(u,v),i(y,v))))).
183 [hyper,1,172,159] P(i(i(i(i(x,y),y),z),i(i(i(u,u),i(v,x)),i(v,z)))).
184 [hyper,1,173,137] P(i(i(x,i(y,z)),i(y,i(i(i(u,u),i(z,v)),i(x,v))))).
185 [hyper,1,106,178] P(i(x,i(i(y,z),i(i(z,i(i(u,u),i(x,v))),i(y,v))))).
186 [hyper,1,181,166] P(i(i(i(i(i(x,x),i(y,z)),z),u),i(i(v,y),i(v,u)))).
187 [hyper,1,4,182] P(i(x,i(i(i(y,y),i(z,i(x,u))),i(i(u,v),i(z,v))))).
188 [hyper,1,182,159] P(i(i(i(x,x),i(y,z)),i(i(i(i(z,u),u),v),i(y,v)))).
189 [hyper,1,182,184] P(i(i(i(x,x),i(y,z)),i(i(i(i(u,y),z),v),i(u,v)))).
190 [hyper,1,172,184] P(i(i(i(i(x,y),z),u),i(i(i(v,v),i(y,z)),i(x,u)))).
191 [hyper,1,106,184] P(i(i(x,i(y,z)),i(i(i(u,u),i(z,v)),i(y,i(x,v))))).
192 [hyper,1,4,184] P(i(x,i(i(y,i(x,z)),i(i(i(u,u),i(z,v)),i(y,v))))).
193 [hyper,1,184,147] P(i(i(x,y),i(i(i(z,z),i(i(i(y,u),u),v)),i(x,v)))).
194 [hyper,1,116,187] P(i(x,i(i(y,i(i(z,z),i(x,u))),i(i(u,v),i(y,v))))).
195 [hyper,1,4,191] P(i(i(i(x,x),i(y,z)),i(i(u,i(v,y)),i(v,i(u,z))))).
196 [hyper,1,191,147] P(i(i(i(x,x),i(i(i(y,z),z),u)),i(i(v,y),i(v,u)))).
197 [hyper,1,106,192] P(i(x,i(i(i(y,y),i(z,u)),i(i(v,i(x,z)),i(v,u))))).
198 [hyper,1,116,193] P(i(i(x,y),i(i(i(i(y,z),z),i(i(u,u),v)),i(x,v)))).
199 [hyper,1,106,194] P(i(x,i(i(y,z),i(i(u,i(i(v,v),i(x,y))),i(u,z))))).

Of the various questions that can be asked, two occupied my attention more than the rest. The first question concerns the—startling to me—small level of two of the proofs cited here, the 48-step proof and the 90-step proof. I have witnessed OTTER's success in producing thousands and thousands of proofs, and, if memory serves, none have had levels of 11 or less when the length has exceeded 45. How could proofs of this type exist? What property could explain the phenomenon?

Overbeek came to my rescue, explaining how a proof could have, for example, length 48 and level 11. The last step, of level 11, could have two parents each of level 10. Each of the two parents could have as parents two formulas of level 9, and, at this point, seven formulas would be accounted for. Then, each of the four formulas of level 9 could have two parents of level 8, eight such formulas. You can continue to outline the nature of the proof. His comments also shows you how a 90-step proof of level 11 could exist. How marvelous and strange!

The second question asks about the existence of a proof, for example, the cited proof of length 90, that is so dense in important information. In particular, as noted, seventy-two of its steps are all of the known single axioms for the *BCI* logic. In other words, only eighteen of the deduced formulas are not among the known single axioms. How could this occur? Ulrich was the rescuer this time. He pointed out that items such as the formula known as *C* transform single axioms into other single axioms. Other formulas in the 90-step proof act sort of like associativity and commutativity, transforming single axioms into single axioms. This phenomenon is, of course, not present in all areas of logic. Further, he told me (or reminded me) that I. Thomas was interested in axiom sets in which most of the formulas considered to be important for the system in question had proofs of low levels; he produced such a set for classical implication, and called it (in the title of his paper) a “nice” axiom set. Ulrich himself found an axiom set for the intuitionistic implicational calculus from which fifteen formulas viewed as important can be deduced in just sixteen steps. So, in a manner of speaking, Halleck and I had stumbled onto an area that had been of interest earlier to various logicians.

So the story ends with an epilogue and implied challenges. Which, if any, of the remaining unclassified formulas is a single axiom? What do the models look like that show that one or more of the remaining is too weak? Can proofs be found that are shorter than those cited here? And, of a different nature—close to Overbeek’s heart—can much of what is offered here be programmed, thus permitting the researcher to spend time on other interesting aspects?