Theo and Octopus at CASC-J3

Monty Newborn
newborn@cs.mcgill.ca

1. Introduction

This paper reports on and analyzes the performances of Theo and Octopus in the 2006 CADE Automated Theorem Proving System Competition held in Seattle on August 18. The event was host by the IJCAR [1]. Twenty programs developed by research groups in Europe, North America, and Australia competed in various divisions. Theo and Octopus participated in the FOF (First-Order Formulas) and CNF (Clause Normal Form) Divisions. Because Octopus ran on non-standard architecture, it participated in the Demonstration System. Each division had 150 theorems to prove, with each theorem allotted 400 seconds. The theorems were taken from the TPTP Problem Library, version 3.2.0 [2]. A number of theorems in the competition were in earlier versions of the library, while a significant number appeared for the first time in the new version. The theorems in the competition were selected randomly from a subset of those in the library. The subset contained theorems that met various criteria such as being not too hard or easy and being spread across different domain areas.

Theo is a resolution-refutation theorem prover [3]. During the competition, it ran on a Dell Precision 330 computer located at the University of Manchester. The computer used an AMD Athlon XP 2200+ processor running at 1797MHz. It had 512Mb of memory and used Linux.

Octopus is a multiprocessing system that runs multiple copies of Theo on as many computers as available. Octopus ran on 133 computers located in various laboratories in McGill University's School of Computer Science. The computers used Intel P4 2.6 Ghz and 3.4Ghz processors and Athlon 64-bit 3.0Ghz processors. Some machines had a gigabyte of memory and others 512Mb. The labs were quiet with students away on summer vacation, giving Octopus exclusive use of most computers. Some computers ran FreeBSD and others Linux.

Theo and Octopus are programmed in C. Each consists of approximately 30000 lines of code. The code transforming first-order formulas to clauses is identical in both programs. Overall, more than 95% of the code in Octopus is identical to that in Theo. Octopus's interprocessor communication routines are written to handle either PVM or MPI. Both programs make use of a large hash table that stores as many as 16M clauses and occupies approximately 200Mb. Another 40 Mb is used by other data structures in the program.

While Theo's performance is of interest in this paper, the performance of Octopus is the focus. Octopus's performance tested a number of ideas related to the use of a learning technique that was implemented several years ago and has been under continual refinement [4–6]. The technique seems to have excellent potential; the degree to which Octopus's performance in the competition supports this contention is examined. Before considering the performances of Theo and Octopus in the competition, it is necessary to provide some background material and describe some features of the programs.

2. The Inferences Performed by Theo and Octopus

A theorem consists of a set of axioms and a conjecture. In the FOF Division, the axioms and conjecture are given as first-order formulas. In the CNF Division, they are given in clause normal form (or conjunctive normal form) with the conjecture negated. When given an FOF theorem, Theo and Octopus negate the conjecture, convert the first-order formulas to clause normal form, and then seek a contradiction by generating inferences. The set of clauses that constitute the theorem are called base clauses. Those that constitute the negated conjecture are said to belong to the set-of-support.

Theo and Octopus perform four types of inferencing operations with no operation involving more than two clauses. They carry out binary resolution, binary factoring, term substitution by a constant or variable, and instantiation of one or two variables in a clause. Most theorem proving programs perform more complex inferencing operations. The operations performed by Theo and Octopus are limited by three parameters: (1) by the maximum number of terms in a literal in the clause, (2) the maximum number of different variables in a clause, and (3) the maximum number of different literals in a clause. Data structures place a limit of 16 on the number of literals in a clause. Theo and Octopus examine a given theorem and attempt to set appropriate values for these three parameters before attempting to find a proof. Octopus differs from Theo in that each slave varies these parameters over a range of values. The net result is that some slaves seek “fat” proofs with literals having many terms or with a large number of literals, while other do the opposite, along with many variations in between.

3. The Learning Strategy: Using Proofs of Weakened Versions

The learning strategy used by Theo and Octopus is a based on the observation that, in general, it is easier to prove some theorem T1 having, say, the conjecture, expressed as a first-order formula:

(1) $\exists x\ P(x)$

than another theorem T2 having the same axioms but, say, having the conjecture:

(2) $\forall x\ P(x)$

That is, it is usually easier to prove some property for one element of some set than for all members. When the technique of proof by contradiction is used, these two conjectures are negated and transformed into clauses:

(1’) ~P(x)

and

(2’) ~P(SK)

where SK is a Skolem constant, and then inferences carried out until a contradiction is found.

Now we maintain that it is generally easier to prove T1 than T2. If we cannot prove T1, we certainly cannot prove T2. This seems obvious enough. However, in practice T1 may turn out to take more time to prove, as T1 will have a larger search space than T2. Note that clause (1’) subsumes clause (2’). By definition, a clause subsumes itself, but we trivially observe here that clause (1’) properly subsumes clause (2’).

Clause Cw is called a weakened version (or a weakening) of clause C if Cw properly subsumes C and Cw is not the null clause. A theorem Tw is a weakened version (or a weakening) of theorem T if each clause in Tw properly subsumes a clause in T. A weakened version of a given theorem is generally easier to prove than the given theorem, though sometimes this is not true as the search space of the weakened version may be larger.

Clause Cw is called a minimally weakened version of clause C if Cw can be derived from C by one of the following four minimal weakening operations:

(1) Replacing a constant by a variable that does not appear in C.
(2) Replacing an n-ary function with n different variables as arguments by a variable that does not appear in C.
(3) Replacing a variable in C with a variable that does not appear in C.
(4) Deleting an n-ary literal with n different variables as arguments.

Note that any weakened version of a theorem T can be derived by a succession of minimal weakening operations. If theorem Tw is a weakened version of theorem T, and if Tww is a weakened version of Tw, then the proof of Tw is usually easier to find than that of T, but more difficult to find than that of Tww. However, the proof of Tw is likely to shed more light on the proof of T than the proof of Tww. In general, it seems the most useful weakenings are those that yield a weakened theorem just a bit weaker than the given theorem. This philosophy is programmed into Theo and Octopus.

4. The Learning Strategy Used by Theo

When Theo attempts to prove a theorem it first tries to prove a weakened version. It allots 50% of the time assigned to prove the given theorem to prove the weakened version. It begins by creating a list called the Weakened Clause List and places on it various ways to weaken individual base clauses. Each way weakens a clause in one of four ways: (1) by replacing a constant by a variable that does not appear in the clause, (2) by replacing a function by a variable that does not appear in the clause, (3) by replacing a variable by another variable that does not appear in the clause, and (4) by deleting a literal. Note that (1) and (3) are minimal weakenings, while (2) and (4) are not.

Various heuristics are used to order the weakenings so that those felt more likely to yield weakened proofs that are effective in proving the given theorem are placed high on the list. Some weakening that are not likely to be useful are eliminated at this point. For example weakening the clause `~equal(a,b)` to `~equal(x,b)` is not likely to be useful, as this clause resolves with `equal(x,x)`, yielding a contradiction in one step, but shedding no information on how to prove the given theorem. Theo then picks the weakening at the top of this list, creates the weakened version of the given theorem, and then attempts to find a weakened proof.

If a proof of the weakened version is found but judged not to be useful for proving the given theorem or if 50% of the allotted time passes and no proof found, another weakened version is attempted. If again this second proof is thought not likely to be useful or if the allotted time runs out, a third weakened version is tired. This time, Theo says to itself that it must use what it can from the proof of this third effort and go on to try to prove the given theorem. If no useful weakened proof is found after three attempts, Theo goes on to prove the given theorem. On its first attempt to find a weakened proof, only one clause is weakened, but on subsequent attempts, two clauses are weakened. There has been some effort to determine the best strategy in this regard.

The proof of a weakened version may, in fact, be a proof of the given theorem. This happens when the weakened clause is not used in the proof of the weakened theorem. If this is the case, Theo gleefully prints out the weakened proof informing the user that a proof of the given theorem has in fact been found. Otherwise, Theo attempts to replicate the proof of the weakened version with the weakened clauses replaced by the original clauses. Again, here, if a proof is found, Theo prints out the results, claiming that the weakened proof yields a proof of the given theorem after certain substitutions are carried out.

If the proof of a weakened theorem doesn't yield a proof of the given theorem, then Theo looks at the length of the weakened proof, and it checks whether the negated conclusion is used in the proof. If these two factors suggest the weakened version might be useful, Theo then gathers what has been learned in proving the weakened version and goes on to attempt the given theorem. In particular, Theo does two things:

> (1) It assigns base clauses used in the proof of the weakened version to the set-of-support. These clauses are searched more deeply that the others. In theorems with a large number of clauses, determining which clauses are relevant to the proof is not easy, and the proof of the weakened version seems quite useful in this regard. A clause used to prove the weakened version has a higher probability of being involved in the proof of the given theorem than other clauses. We look at the effectiveness of this in Section 8.2.
>
> (2) Some inferences in the proof of the weakened version that follow when the original clauses replace the weakened clauses and the same inferences attempted are added to the base clauses of the given theorem and assigned to the set-of-support. There are various heuristics that decide which inferences to add. The added clauses are instances of clauses that yielded the weakened proof, and experience has shown that they are helpful in proving the given theorem. The effectiveness of this is considered in Section 8.3.

The proof of the given theorem is sought in the remaining time. Theo may now find a proof of the given theorem exists, but that certain inferences whose derivations were not saved must be derived again to piece together a proof. Finding the derivation of these inferences may require backtracking to find still other inferences and their derivations. Theo will go on to take whatever time is necessary to construct the proof whether or not it exceeds the time bound. During the competition, this case did not occur for Theo. That is, if Theo found a proof exists in less than 400 seconds, it also completed the construction of the proof in less than 400 seconds. If Theo had required more than the 400 seconds, the shell program under which the program ran would have cut off the attempted construction. Octopus, on the FOF theorems found several proofs in less than 400 seconds, but took more than 400 seconds to complete their construction.

5. Octopus and its Differences with Theo

The master and each slave of Octopus runs a copy of Theo. The master begins by reading in the theorem from disk, sending it to the slaves, and then proceeding to search for a proof. During the search, it also monitors messages from the slaves regarding their progress. Rather than selecting the first weakening on the Weakened Clause List as does Theo, each of Octopus's slaves picks a different weakening. In particular, the i-th slave selects the i-th weakening. Actually, it's not quite that simple. Because the weakenings are ordered by various heuristics to place those potentially more effective near the top of the list, the first approximately 25% of the even numbered slaves pick the same weakening as the preceding odd numbered slave. That is, for the first 25% of the slaves, the i-th weakening is assigned to slave 2i – 1 and slave 2i. The remaining slaves pick consecutive weakenings on the list.

Unlike Theo, Octopus's slaves try one weakening and spend the entire allotted time trying to prove the corresponding weakened theorem. However, if a proof of the weakened theorem is found but judged not to be useful, a second attempt is made weakening another base clause. If a weakened proof is found the first time or any proof found the second time, the slave then attempts to prove the given theorem.

Otherwise, the procedure carried out by each slave is the same as Theo — with several exceptions! First, each slave uses different limits on the number of literals and variables in a clause and the number of terms in a literal. Thus, two consecutively-number slaves might pick the same weakening but search for a proof using different clause limits. Unlike the slaves, the master attempts to prove the theorem without weakening it. For simple theorems, the master quickly finds a proof; the results show that the master proved many theorems. This happens less often when the time to prove a theorem takes more than a few seconds.

Octopus also uses splitting on a rather restricted basis. In particular, splitting is only carried out on base clauses. More than one base clause can be split, but a limit of 16 sub-theorems is set. When a slave proves a split sub-theorem, it reports the results to the master. It then goes on to attempt the next split sub-theorem, reporting its proof if found to the master, and then going on to the next split sub-theorem as time permits. Thus one slave may prove a number of the split sub-theorems.

Once a slave finds a proof exists and no splitting is involved, it signals the master of this, and the master kills the other slaves. Octopus could have been programmed not to kill the slaves, but this was done to keep matters simple. This means that sometimes another slave might have found a proof exists later than the one signaling the master, but have completed the construction sooner. This case is lost. If the master finds a proof exists, it doesn't kill the slaves. A slave may find a proof exists after the master, but completes the construction sooner. All a bit irregular, but that's the way Octopus is programmed. If splitting is involved, no slave is killed until proofs have been received for all sub-theorems.

6. Theo and Octopus in the 2006 CASC ATP System Competition

The competition began at about 10AM PST. Theo's FOF and CNF theorems were proved on the Manchester computers along with the other entrants in these two divisions. Detailed results are available on the CASC-J3 website [7]; each theorem attempted by Theo as well as the other entrants has a link to the output produced by the program. The entrants are given credit for finding that a proof of a theorem exists and credit for actually constructing the proof. Five entrants, E.99, Darwin 1.3, Muscadet 2.6, iProver 0.1, and Equinox 1.0a, find only that a proof exists, and do not provide a proof. Vampire 8.0 won last year's competition and was in the competition as a marker of progress by the entrants. Otter 3.3 competed in the 2004 and 2005 competition as might be considered a second marker.

In the FOF Division, Theo found proofs of 94 theorems and constructed proofs of all of them. It finished fifth, behind (with the number of proofs found to exist and the number of proofs constructed in parentheses and separated by a slash) Vampire 8.1 (128/128), E.99 (112/0), EP.99 (112/92), and Vampire

8.0 (109/109). Following Theo were Geo 2006i (73/73), Darwin 1.3 (64/0), Otter 3.3 (52/52), Muscadet 2.6 (50/0), Equinox 1.0a (31/0), and Faust 1.0 (27/27).

In the MIX Division, Theo found proofs of 66 theorems and constructed proofs of them all. It finished fifth, again, behind Vampire 8.1 (141/141), Vampire 8.0 (138/138), E (126/0), and EP (125/103). Following Theo were Otter 3.3 (55/55), Darwin 1.3 (47/0), iProver (47/0), Geo2006i (45/45), Faust1.0 (19/19, and Equinox 1.0a (12/0).

Octopus attempted the same theorems as Theo, though off-line on the McGill computers. Geoff Sutcliffe, in charge of the competition, downloaded the theorems to the McGill system shortly after the other entrants had started. Preparing the theorems for Octopus took several hours, and attempts to prove them began around 2PM that day. Forty-eight hours later Octopus was put to rest. The detailed results in of the competition in the FOF and CNF Divisions that appear on the CASC-J3 website have been modified to include Octopus's results and are presented in Appendix 1. Several entries that finished behind Theo do not appear for space considerations. Entries for Octopus of the form x/y specify that the program found a proof exists in x seconds and constructed it in y seconds. Theorems where Octopus failed to complete the fof->cnf transformation are marked with a "Failed!!!" and placed at the end of the CNF Division list; "Failed!!" denotes the theorem had clauses with too many literals for Octopus to handle.

Before the competition started and after every fifteen theorems, Octopus was programmed to "clean" the network by deleting various files and restarting PVM. The cleaning took about 10-15 minutes each time, adding several hours to the time taken by the program. Each theorem was allotted 400 seconds, the same time limit used in the main competition. The program ran from an Apple MacBook. For the first day, the laptop was in the Competition Room. For the second day, it was located in my hotel room. During this period, the line would go dead if inactive for more than several minutes causing a certain amount of frustration and constant attention to the laptop.

Octopus attempted theorems in the FOF category first. Upon completion, Octopus found that proofs exist to 110 theorems and constructed proofs to 106 of them. This is 16 more than the 94 that Theo found to have proofs and for which proofs were constructed. The four theorems for which Octopus found proofs exist but didn't complete the construction in the 400-second period were all eventually proved.

The strength of the performance raised suspicions. One should always be suspicious when a program performs better than expected! Octopus essentially did one better on the FOF theorems than did Vampire 8.0. Upon analysis of the results, one theorem, `SEU075+1`, was found to have an erroneous proof. In essence, the code that determined whether a weakened proof is in fact a proof of the theorem had a bug. The bug affected only this theorem and was obvious upon examining the results. The bug was present in Theo, but did not come up during the competition because Theo didn't happen upon the weakened proof that Octopus did. The bug was fixed before Octopus began the CNF theorems. Octopus went on to prove 83 CNF theorems, finding proofs to all 83. This compares with Theo's 66. After completing the CNF theorems, the FOF theorems were run a second time. This time, proofs were found to exist to 108, and, proofs to 107 were constructed. Octopus found a proof exists for `AGT018+1` in 383 seconds on the first run, but failed to do so on the second run. All in all, Octopus attempted to prove 450x132 theorems, about 60,000 theorems for round numbers, during the competition and one bug surfaced. In some sense, this might be viewed positively!

Octopus failed when reading in 16 theorems in the CNF Division. These theorems were all large containing thousands of clauses; each occupied more then 230Kb on disk. These theorems had not appeared in previous versions of the TPTP Problem Library. Meng and Paulson had carried out experiments on them using Vampire and E in the last year [8]. They had developed heuristics for choosing clauses that were likely to be more relevant than others when trying to prove these theorems. The heuristics were evidently quite effective, as Vampire and E proved most of these theorems, while the other entrants, including Theo and Octopus, did poorly.

Theo and Octopus failed on the CNF theorem `NLP080-1` because neither program can handle a theorem with more than 16 literals in a clause. To correct this shortcoming requires considerable effort. Theo and Octopus are unable to prove many theorems in the `NLP` domain for this reason.

In the FOF Division, Theo and Octopus failed on five theorems when transforming first-order formulas to clauses. This was due to generating too many clauses and due to shortcomings in the fof->cnf converter.

Theo and Octopus both did quite poorly on theorems that others proved using paramodulation or variations of it. These were theorems in the `GRP` and `LAT` domains, and to a lesser extent the `LCL` domain.

Of the 37 theorems in these three domains of the FOF division, Octopus proved only eleven. The program did prove three of three in these domains in the MIX division.

So where did Theo and Octopus, in particular, do well? Of the 300 theorems, there was only one that Theo and Octopus proved but baffled the other entrants. That was `SET919+1`. Theo's 21-second proof in which a weakened proof was first found is presented in Appendix 2. Octopus proved the theorem on the master, and the learning strategy was not involved. The use of hash tables by Octopus and Theo was instrumental in these two programs finding proofs.

Theo did much better in the FOF Division than in the CNF Division, being plagued by the problems discussed above in the latter division. In the FOF Division, Theo finished 21 theorems better than Geo, the sixth place finishers, while 15 theorems behind Vampire 8.0. In the CNF Division, Theo finished well behind the leading four entries, but clearly ahead of the sixth place finisher Otter 3.3 with 55 theorems proved.

In the FOF Division Octopus outscored Theo by 14 theorems. In ten of the sixteen domains from which theorems were selected in the FOF Division (`AGT`, `ALG`, `GEO`, `GRP`, `LCL`, `MGT`, `MSC`, `NUM`, `PUZ`, and `SYN`), Octopus failed proving all or all but one of those attempted (although the `MSC` domain only had one theorem). Octopus proved 36 of the 40 theorems in these domains. In the `SET` domain, Octopus did very well, solving 43 of 52 theorems, outdoing Vampire 8.1 on these theorems. Of these 92 theorems, Octopus proved one fewer than Vampire 8.1. In the other five domains with 58 theorems (`CSR`, `GRA`, `SEU`, `SWC`, and `SWV`), Octopus fell short of Vampire 8.1's performance by 19 theorems.

In the CNF Division, Octopus outscored Theo by 17 proofs. Excluding the sixteen theorems on which the program failed when reading them into memory, the one that had too many literals in its clauses, and the twenty in the `GRP` and `LAT` domains, Octopus proved 78 of the remaining 112, still clearly short of the 91 proved on the same theorems by the fourth place finisher EP.99.

Three Results Tables summarizing various factors related to the proofs found by Octopus for the FOF (Run 1), FOF (Run 2), and the CNF theorems are listed respectively in Appendices 3, 4, and 5. Each theorem occupies one row if spitting was not involved. If a K-way split was carried out, there are K rows of data reporting on the proof of each split sub-theorem. The theorem number appears in the first column, indicating the order in which theorems were proved during the competition. The second column gives the name of the theorem. The third column specifies the computer that found and constructed the proof. The fourth column gives the number of the weakened versions that Octopus place on the Weakened Clause List (a maximum of 2000 weakened clauses cab be placed on the list). The fifth column specifies the number of the weakening selected, with the number preceded by a dash if the weakening was the second one tried. The number of base clauses in the theorem is given next. If splitting was involved, the split sub-theorem proved, followed by the number of splits is given next. This is follow by the values of the limits on literals, variables, and terms. The twelfth column gives the time taken to find that a proof exists as recorded by the master. The thirteenth column gives the time to complete the construction. The times in columns eleven and twelve are often several seconds more than the times recorded in the file produced by the slave that finds a proof due to delays in the master obtaining and recording the proof from the slave. The last column gives the length of the proof. These tables along with the proofs themselves are used to analyze the performance of Octopus.

7. The Differences between Octopus's Performance on FOF (Run 1) and FOF (Run 2)

As said previously, when the bug was found and fixed, a decision was made to rerun the FOF theorems. In this section, we examine the differences between the two runs (Table 1). The proofs of 72 theorems were identical and were produced by the same computer in both runs. Fifteen of the other 37 were identical but found by different computers. Nine of the remaining 22 involved splitting, whereby different computers found proofs; the odds are very high that different proofs will be found in these cases.

Of the thirteen remaining theorems, the files produced by the slaves and the information in Results Tables (Run 1) and Results Table (Run 2) indicate that proofs were found to exist for five theorems `CSR019+1`, `NUM398+1`, `SET008+3`, `SET579+3`, and `SET595+3`) in both runs in less than a second, although the master sometimes recorded it took several more seconds. The race for attention on the network often leads to different slaves receiving credit for a proof. Seven theorems remain. Of these, four theorems (`SET067+1`, `SET144+3`, `SET577+3`, and `SET935+1`) were proved in no more than eight seconds; this is not unusual for the same reason as for the previous five, and that the computers with 1Gb of memory often find proofs faster than those with 512Mb. SET812+4, taking 29 seconds on slave 54 in Run 1 and 38 seconds in Run 2 on slave 5, is most likely an example of different computers proving theorems at different speeds. In Run 2, two of the remaining three theorems, AGT007+2 and GRA007+1, both happened to have proofs found to exist more quickly and the actual proofs constructed within the 400-second time limit. In Run 1, although proofs were found to exist in the allotted time, their constructions exceeded it. For AGT018+1, Octopus managed to find a proof exists in Run 1 in the allotted time, though failing to construct it during that period; it failed even to find a proof exists in the allotted time in Run 2.

Theorem	Analysis of difference in proofs between Run 1 and Run 2
AGT007+2	Proved on computers with different speeds
AGT018+1	Proved on computers with different speeds
CSR019+1	Proved quickly in both runs
GEO085+1	Same proof found but by different computers
GRA004+1	9-way split; different computers contributed to proof
GRA007+1	Proved on computers with different speeds
LCL414+1	Same proof found but by different computers
MGT042+1	16-way split; different computers contributed to proof
NUM398+1	Proved quickly in both runs
SET008+3	Proved quickly in both runs
SET009+3	Same proof found but by different computers
SET067+1	Proved on computers with different speeds
SET144+3	Proved on computers with different speeds
SET577+3	Proved on computers with different speeds
SET579+3	Proved quickly in both runs
SET584+3	Same proof found but by different computers
SET587+3	4-way split; slaves in Run 1, M in Run 2
SET589+3	Same proof found but by different computers
SET594+3	Same proof found but by different computer
SET595+3	Proved quickly in both runs
SET603+3	Same proof found but by different computers
SET624+3	12-way split; slaves in Run 1, M in Run 2
SET655+3	Same proof found but by different computers
SET657+3	Same proof found but by different computers
SET669+3	2-way split; M proved in Run 1; slaves in Run 2
SET798+4	Same proof found but by different computers
SET802+4	12-way split; different computers contributed to proof
SET812+4	Proved on computers with different speeds
SET935+1	Proved on computers with different speeds
SET966+1	Same proof found but by different computers
SEU003+1	Same proof found but by different computers
SEU009+1	12-way split; M proved in Run 1; slaves in Run 2
SWC239+1	16-way split; different computers contributed to proof
SWV030+1	16-way split; different computers contributed to proof
SWV041+1	Same proof found but by different computers
SWV235+1	Same proof found but by different computers
SYN365+1	Same proof found but by different computers

Table 1. Thirty-seven theorems for which Octopus differed in its proofs in Run 1 and Run 2.

8. Analysis of the Results

In this section, we consider five aspects of Octopus's performance. The first is a comparison between it and Theo regarding the number of theorems proved. The second is a count of the contribution of the i-th computer to the proofs found. One would hope the data shows a good distribution, implying that the technique has the potential to extend effectively to even larger numbers of computers. The third is an examination of the effectiveness of assigning base clauses used in a weakened proof to the set-of-support. One would hope that a high percentage of those assigned were used in the proof of the given theorem. The fourth is an examination of the effectiveness of adding inferences used in a proof of the weakened version to the base clauses and placing them in the set-of-support. Here again, one would hope that a high percentage of these added clauses are used in the proof of the given theorem. The fifth is a comparison of the Octopus's performance in Seattle with an identical version run on the same theorems two weeks later on the same system but with no learning. One would expect that Octopus with no learning outperformed Theo, but fell short of Octopus when using the learning strategy.

8.1 A Comparison of the Performance of Octopus and Theo

For the FOF theorems, a comparison is based on the second run data. First, every theorem proved by Theo was also proved by Octopus. This is not necessarily always the case as minor differences in the learning strategy and clause limits result in searches by Octopus that do not necessarily subsume those by Theo.

In the FOF category, Octopus found proofs exist for 108 theorems and constructed proofs to 106 of them; Theo found proofs exist and constructed proofs for 94. In the CNF category, Octopus found proofs exist and constructed proofs of 83 theorems; Theo found proofs exist and constructed proofs for 66. It is interesting to examine and compare the time Octopus spent on theorems that took Theo more than 30 seconds to find and then construct a proof: There are 24 cases and they are given in Table 2a (where the asterisk indicates Octopus split the theorem and the length of the proof is the sum of the split proofs). We see that Octopus proved these 24 theorems on average 21.2 times as fast as Theo did.

Theorem	Theo (Seconds)	Octopus (Seconds)	Octopus (Length)	Speedup
AGT024+2	264	93	43	2.8
ANA002-1	231	7	69	33.0
ANA003-2	228	10	25	22.8
ANA004-5	279	4	30	69.7
COM009-1	64	23	6	2.7
GEO007-1	308	19	74	16.2
GEO011-1	83	4	32	20.7
GRP039-5	274	45	114	6.0
HWV016-1	109	7	89	15.5
LCL237-3	86	6	21	14.3
MGT042+1	131	12	*439	10.9
NUM010-1	125	37	51	3.3
NUM017-1	88	1	22	88.0
NUM400+1	53	47	52	1.1
PLA014-1	58	1	39	58.0
SET013-1	125	3	19	41.6
SET171+3	42	20	80	2.1
SET208-6	214	23	23	9.3
SET841-1	159	7	17	22.7
SWC239+1	159	39	*205	4.0
SWV030+1	376	15	*218	25.0
SWV048+1	124	7	19	17.7
SWV098+1	64	8	*24	8.0
SWV242-1	162	12	10	13.5
			Average speedup	21.2

Table 2a. Data comparing Theo and Octopus on 24 theorems that took Theo more than 30 seconds to find a proof exists and then to construct it.

In addition to these 24, Octopus found proofs exist to 31 other theorems and constructed proofs to 30 of them, and Theo failed on them all. The 31 theorems are listed in Table 2b along with Octopus's time to find a proof exists, the proof length, and the speedup. The asterisk means the same here as in Table 2a. The speedup for each theorem is measured assuming Theo took 400 seconds. This is a very hypothetical calculation; Theo may never have been able to prove some of them, as its search strategy is incomplete. If one assumes that Theo proved each of them in 400 seconds, a lower bound on the time Theo might have required, Octopus effectively proved these 31 theorems on average at least 51.4 times as fast as Theo might have if given more time. The results, with the two best and worst speedup removed, lead to a speedup of 23.2. The speedup of Octopus over Theo on these theorems is impressive.

Theorem	Octopus (Seconds)	Octopus (Length)	Average Speedup Octopus vs. Theo
AGT007+2	154	43	2.5
ALG074+1	3	73	133.3
COL003-2	51	25	7.8
COL093-2	114	*33	3.5
CSR007+1	110	80	3.6
GEO076-4	8	77	50.0
GEO088+1	4	35	100.0
GEO089-1	36	32	11.1
GRA007+1	302	104	1.3
GRA007+2	270	105	1.4
GRP311-1	249	*1624	1.6
PLA010-1	47	42	8.5
PLA016-1	1	22	400.0
PLA018-1	46	60	8.6
SET018-3	63	19	6.3
SET028-3	15	29	26.6
SET582+3	7	62	57.1
SET787-2	16	13	25.0
SET795+4	18	22	22.2
SET802+4	20	*58	20.0
SET811+4	24	26	16.6
SET812+4	17	23	23.5
SET831-2	92	*234	4.3
SET853-1	22	45	18.1
SET951+1	30	50	13.3
SEU003+1	78	24	5.1
SWV050+1	159	19	2.5
SWV248-2	1	20	400.0
SWV282-2	3	28	133.3
SYN014-1	5	*124	80.0
SYN036-3	52	*391	7.6
Average speedup			51.4
Ave. speedup w/o 2 best/worst cases			23.3

Table 2b. Data on thirty-one theorems that Theo was unable to prove but that Octopus was able.

8.2 Scaling

The 55 theorems from the previous section are now used for further analysis of the effectiveness of Octopus's learning strategy. In this section, the issue of scalability is considered. Table 3 gives data on which computer proved each of the 55 theorems. In cases where splitting was involved, all computers that proved sub-theorems are listed. `M` denotes the theorem was proved by the master. The data is plotted in

Table 4, and it shows that approximately 28% of the theorems were proved by computers numbered over 60. Computers numbered between 121 and 132 participated in the proofs of six theorems.

Theorem	Number of computer that found proof (M = Master)				
AGT024+2	M	NUM400+1	M	SET171+3	57
SWV048+1	114	AGT007+2	44	ALG074+1	79
CSR007+1	52	GEO088+1	3	GRA007+1	90
GRA007+2	66	SET582+3	56	SET795+4	26
SET811+4	104	SET812+4	5	SET951+1	132
SEU003+1	3	SWV050+1	13		
MGT042+1	4,6,10,2,14,12,9,1,10,2,4,13,111,23,111,9				
SET802+4	2,4,3,6,8,4,7,10,8,1,96,95				
SWC239+1	56,49,9,66,56,49,56,56,49,56,49,56,54,49,72				
SWV030+1	6,26,4,41,30,2,49,4,26,6,76,30,34,6,76,26				
SWV098+1	24,24				
ANA001-1	23	ANA003-2	63	ANA004-5	54
COM009-1	M	GEO007-1	18	GEO011-1	89
GRP039-5	14	HWV016-1	21	LCL237-3	35
NUM010-1	60	NUM017-1	64	PLA014-1	11
SET013-1	14	SET208-6	4	SET841-1	M
SWV242-1	M	COL003-2	107	GEO076-4	94
GEO089-1	110	PLA010-1	6	PLA016-1	1
PLA018-1	43	SET018-3	72	SET028-3	123
SET853-1	26	SWV282-2	102		
COL093-2	119,128,122,2				
GRP311-1	5,4,56,36,16,56,16,21,16,24,95,26,26,104				
SET831-2	29,30,28,32,34,33,6,31,36,40,35,42,104,117,86,27				
SYN014-1	33,64				
SYN036-3	91,87,78,94,81,108,65,121,98,116,95,123				

Table 3. List of processors that proved 55 theorems of interest.

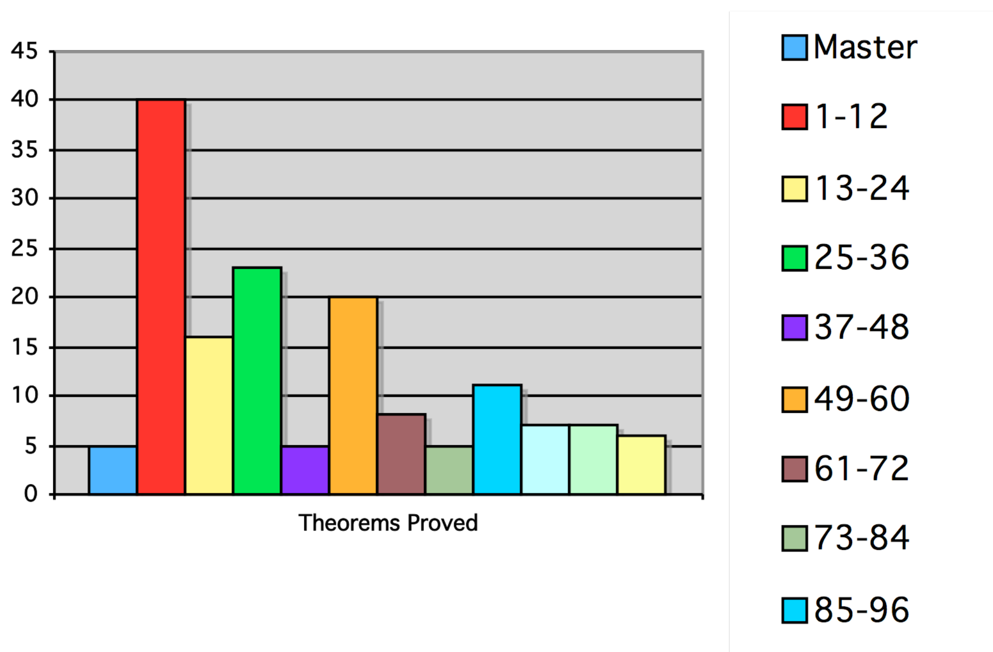

Table 4. The number of theorems proved as a function of the computer number grouped by each twelve consecutive computers.

8.3 The Effectiveness of Assigning Axioms Used in Weakened Proofs to the Set-of-Support

To evaluate the effectiveness of assigning axioms used in weakened proofs to the set-of-support, we consider the same 55 theorems listed anew in Table 5. The table lists the names of the theorems in the first column. The next five columns give: the number of base clauses, the length of the weakened proof, the length of the proof of the given theorem, the time to prove the weakened theorem, and the time to find a proof exists of the given theorem. The next column specifies the type of weakening that led to the proof. The last four columns specify: the number of clauses assigned to the set-of-support, the number and percent of those added and used in the proof of the given theorem, the number of axioms in the theorem, and the number and percent of axioms used in the proof of the given theorem.

Theorems that were proved by the master (denoted by an `M`) and those whose proof followed directly from the weakened proof (denoted by `WPS`) are listed at the end of the table. Data is not provided in these seven cases, as learning was not used. Data is also not provided for the ten theorems where splitting (denoted by `SPLIT`) was used to find a proof. Thirty-eight theorems remain for which data is presented.

The most significant conclusion is that 58% of the axioms assigned to the set-of-support by the learning strategy were using to prove the given theorem. This contrasts with 38% of the axioms used more generally. For theorems with more than 100 base clauses, the numbers are more striking: 54.7% versus 8.4%. Thus, the learning strategy was effective in assigning promising axioms to the set-of-support.

In nine of the 38 proofs, the weakened clause was created by replacing a constant by a variable not in the clause. In twelve cases, it was created by replacing a function by a variable not in the clause. In ten cases, it was created by replacing a variable by another variable not in the clause. Lastly, in seven cases, it was created by deleting a literal. This shows the diversity of weakenings that are successful.

In only six cases (`CSR007+1`, `GRA007+1`, `GEO089-1`, `SET795+4`, `SEU003+1`, and `SWV050+1`) did the weakened proof take more than 10 seconds to find. In Section 8.4 (See Table 7), it is shown that a 132-slave Octopus system that didn't use the learning strategy was able to prove only one (`SEU003+1`) of these six, strongly suggesting that the learning was crucial in these cases. In three of these six, the proof of the given theorem was found in no more than one second after the weakened proof was obtained.

On average, the length of a weakened proof was approximately two-thirds the length of the proof of the corresponding given theorem. In eight cases, the proof of the weakened version was longer than the proof of the given theorem.

Theorem	Base	LW	LP	TW	TT	Type	Assigned to SOS	Assigned Used(%)	Axioms	Axioms Used(%)
AGT007+2	1004	7	43	2	142	f->v	0	0(00.0)	1003	30(3.0)
ALG074+1	25	74	47	2	0	f->v	14	8(57.1)	19	9(47.4)
ANA002-1	18	43	59	1	2	del_lit	13	10(76.9)	17	14(82.4)
ANA003-2	17	6	25	0	9	f->v	2	1(50.0)	10	5(50.0)
ANA004-5	16	16	22	1	1	f->v	10	5(50.0)	15	6(40.0)
COL003-2	10	15	18	1	92	v->v	7	5(71.4)	9	6(66.7)
CSR007+1	202	40	66	49	61	f->v	21	11(52.4)	201	37(18.4)
GEO007-1	60	57	74	2	17	v->v	20	15(75.0)	59	19(32.2)
GEO011-1	63	29	27	2	1	v->v	12	8(66.7)	62	9(14.5)
GEO076-4	60	17	76	1	7	v->v	7	4(57.1)	59	13(22.0)
GEO088+1	90	15	31	1	3	c->v	5	5(100.0)	86	10(11.6)
GEO089-1	89	49	17	34	1	v->v	13	3(23.1)	86	4(4.7)
GRA007+1	123	55	86	231	71	f->v	16	6(37.5)	119	17(14.3)
GRA007+2	124	3	104	0	270	del_lit	2	2(100.0)	120	21(17.5)
GRP039-5	23	42	120	2	43	c->v	11	11(100.0)	19	16(84.2)
HWV016-1	125	45	67	2	4	c->v	16	9(56.3)	120	19(15.8)
LCL237-3	22	7	20	0	4	v->v	5	2(40.0)	21	11(52.4)
NUM010-1	407	16	51	5	37	del_lit	8	6(75.0)	406	17(4.2)
NUM017-1	24	9	19	0	0	v->v	5	1(20.0)	23	6(26.1)
PLA010-1	31	36	32	8	39	c->v	22	13(59.1)	30	17(56.7)
PLA014-1	31	36	14	0	0	v->v	16	5(31.3)	30	7(23.3)
PLA016-1	31	8	20	0	0	c->v	8	7(87.5)	30	15(50.0)
PLA018-1	31	27	52	1	74	v->v	15	8(53.3)	30	20(66.7)

SET013-1	22	9	17	1	1	c->v	5	5(100.0)	21	6(28.6)
SET018-3	274	4	19	2	61	f->v	3	2(66.7)	273	12(4.4)
SET028-3	279	4	29	0	12	del_lit	3	3(100.0)	278	16(5.8)
SET171+3	35	11	80	0	20	f->v	5	2(40.0)	34	11(32.4)
SET208-6	189	0	16	3	23	c->v	2	0(0.0)	187	10(5.3)
SET582+3	46	10	65	0	7	del_lit	4	3(75.0)	41	13(31.7)
SET795+4	241	12	18	17	1	c->v	9	1(11.1)	238	9(3.8)
SET811+4	144	4	45	4	26	f->v	3	3(100.0)	141	12(8.5)
SET812+4	143	4	23	0	38	c->v	3	3(100.0)	141	10(7.1)
SET951+1	64	15	46	0	40	f->v	9	5(56)	62	10(16.1)
SEU003+1	97	24	20	76	1	c->v	11	6(55.6)	91	5(5.5)
SWV048+1	173	9	19	1	3	v->v	5	2(40.0)	167	11(6.6)
SWV050+1	164	19	11	55	103	f->v	11	3(27.3)	160	5(3.1)
SWV248-2	9	7	16	0	0	del_lit	5	3(60.0)	8	7(87.5)
SWV282-2	42	19	22	1	1	f->v	10	4(40.0)	41	5(12.2)
AGT024+2	M									
COL093-2	SPLIT									
COM009-1	M									
GRP311-1	SPLIT									
MGT042+1	SPLIT									
NUM400+1	M									
SET787-2	WPS									
SET802+4	SPLIT									
SET831-2	SPLIT									
SET841-1	M									
SET853-1	WPS									
SWC239+1	SPLIT									
SWV030+1	SPLIT									
SWV098+1	SPLIT									
SWV242-1	M									
SYN014-1	SPLIT									
SYN036-3	SPLIT									
							% Used	58.1	% Used	27.9

Table 5. A summary of axioms placed in the set-of-support and used versus all axioms and those used for theorems proved by Octopus 25 times faster than by Theo or by Octopus in over one minute on the master.

8.4 The Effectiveness of Adding Inferences Used in the Weakened Proof to the Base Clauses and Assigning Them to the Set-of-Support

For this section, the same 38 theorems for which data was presented in the previous section are considered again. The theorems are listed in Table 6. The name of each theorem appears in the first column. The length of the weakened proof is given in the second column. The number of inferences added to the base clauses and assigned to the set-of-support is given in the third column. The fourth column gives the number and percentage of those added and used in the proof of the given theorem. The fifth column gives the number of base clauses in the theorem. The sixth column gives the number and percentage of base clauses used in the proof of the given theorem. The data in columns 2 and 3 show that 36.2% of the inferences in the proofs of the weakened versions were added to the base clauses to help prove the given theorem. The weakened proof of `ALG074+1` was 74 inferences long, and 25 of them were added to the base clauses, very near the average.

The data in columns 3 and 4 show that 36.2% of the inferences in the proofs added to the base clauses and placed in the set-of-support were used in the proofs. Columns 5 and 6 show that this compares with 28.4% of the given base clauses that were used. For theorems with over 100 axioms, 28.8% of the inferences added and placed in the set-of-support were using in the proof of the given theorem. This compare with 8.7% of the base clauses used. One can concluded that the learning strategy is fairly effective in selecting inferences from the weakened proof to add to the base clauses and assign to the set-of-support.

Theorem	WP Length	NEW Base	New Used(%)	Orig Base	Orig Used(%)
AGT007+2	7	0	0(0.0)	1004	31(3.1)
ALG074+1	74	25	12(48.0)	25	10(40.0)
ANA002-1	43	4	2(50.0)	18	14(77.8)
ANA003-2	6	4	1(25.0)	17	11(64.7)
ANA004-5	16	7	5(71.4)	16	7(43.8)
COL003-2	15	5	2(40.0)	10	6(60.0)
CSR007+1	40	19	6(31.6)	202	30(14.9)
GEO007-1	57	0	0(0.0)	60	20(33.3)
GEO011-1	29	10	5(50.0)	63	10(15.9)
GEO076-4	17	6	3(50.0)	60	14(23.3)
GEO088+1	15	6	2(33.3)	90	14(15.6)
GEO089-1	49	13	4(30.8)	89	6(6.7)
GRA007+1	55	13	7(53.8)	123	20(16.3)
GRA007+2	3	2	2(100.0)	124	24(19.4)
GRP039-5	42	9	4(44.4)	23	20(87.0)
HWV016-1	45	20	8(40.0)	125	21(16.8)
LCL237-3	7	5	1(20.0)	22	11(50.0)
NUM010-1	16	4	0(0.0)	407	18(4.4)
NUM017-1	9	4	2(50.0)	24	6(25.0)
PLA010-1	36	15	5(33.3)	31	17(54.8)
PLA014-1	36	14	4(28.6)	31	7(22.6)
PLA016-1	8	3	1(33.3)	31	15(48.4)
PLA018-1	27	10	7(70.0)	31	20(64.5)
SET013-1	9	4	1(25.0)	22	7(31.8)
SET018-3	4	0	0(0.0)	274	5(1.8)
SET028-3	4	2	0(0.0)	274	16(5.8)
SET171+3	11	5	2(40.0)	35	12(34.3)
SET208-6	3	2	2(100.0)	189	11(5.6)
SET582+3	10	2	1(50.0)	46	18(39.1)
SET795+4	12	2	1(50.0)	241	12(5.0)
SET811+4	4	1	0(0.0)	144	15(10.4)
SET812+4	4	1	0(0.0)	143	12(8.4)
SET951+1	15	8	3(37.5)	64	12(18.8)
SEU003+1	24	10	3(30.0)	97	9(9.3)
SWV048+1	9	3	0(0.0)	173	12(6.9)
SWV050+1	19	7	2(28.6)	164	6(3.7)
SWV248-2	7	4	2(50.0)	9	7(77.8)
SWV282-2	19	8	5(62.5)	42	6(14.3)
		% Used	36.2	% Used	28.4

Table 6. Data on the number of new base clauses add to the given theorem and placed in the set-of-support.

8.5. Theo and Octopus Using Learning Versus Theo and Octopus Not Using Learning

In the days following the competition, Theo and Octopus attempted the same FOF and CNF theorems but without using the learning strategy. The versions of Theo and Octopus that don't use the learning strategy are called TheoNL and OctopusNL, respectively. Here we compare the results with those obtained in Seattle. TheoNL found proofs exist and constructed proofs of 65 CNF theorems. It found proofs exists of 97 FOF theorems and constructed proofs of 96 of them. In total, it found proofs exist of three more theorems than Theo and constructed proofs of two more theorems.

Speed tests were subsequently carried out on the McGill computer used by TheoNL, and it was found that it was about 70% faster than the Manchester computers used in the competition. This also can be seen by examining the results of several theorems proved by both with and without learning. `GEO011-1` and `SWV048-1` illustrate this best. Both arrived at essentially the same proofs while carrying out very similar searches. For `SWV048-1`, 675K nodes in the search space on the Manchester computer were examined in 124 seconds, while 554K nodes in the search space on the McGill computer were examined in 56 seconds. To equalize the two computers, one might have limited the time per theorem on the McGill computer to

250 seconds or so. If this had been done, TheoNL would have found proofs to three fewer theorems and constructed proofs to four fewer. The two performances would have resulted in an identical number of proofs found to exist by both programs, but one more proof constructed by Theo.

Essentially, the learning strategy barely improved Theo! In many cases, using the weakened proof made finding the real proof more difficult by enlarging the search space. This was also the case for Octopus. However, Octopus created many weakened versions, and sometimes one of them paid off.

Table 7 (See also Appendix 6) presents the results of testing OctopusNL on the 31 theorems that Octopus proved in Seattle but that Theo didn't. Of these, sixteen — over half — went unproven by OctopusNL. Of the fifteen others, Octopus proved five of them at least five times faster than OctopusNL. It proved twelve of the sixteen at least twice as fast. Only in two cases did it take more time when finding a proof. Assigning 400 seconds to the time taken by OctopusNL on the theorems it failed to prove, the speedup of Octopus over OctopusNL is 14.93. Eliminating the two extreme high and low speedups, the speedup is 10.96.

The average computer in Octopus's network was faster than the Manchester computer as said previously. Given this difference and given that Octopus took more than 250 seconds to find proofs exist to two theorems, one might contend that Octopus proved only 29 more theorems than did Theo.

A similar experiment was carried out several years ago to test Octopus versus OctopusNL [4]. Proofs of 43 1.0-rated theorems were reported found by Octopus using an early version of the learning strategy. OctopusNL found proofs of 29. In this most recent test, the percent proved by OctopusNL on a difficult set of theorems was somewhat lower than on the earlier test, perhaps indicating that the learning strategy had become somewhat more effective.

Theorem	Octopus(Time)	OctopusNL(Time)	Speedup
AGT007+2	154	269	1.75
ALG074+1	3	47	15.67
COL003-2	51	Failed	7.84
COL093-2	114	Failed	3.51
CSR007+1	110	Failed	3.64
GEO076-4	8	Failed	50.00
GEO088+1	4	Failed	100.00
GEO089-1	36	Failed	11.11
GRA007+1	302	Failed	1.32
GRA007+2	270	Failed	1.48
GRP311-1	249	Failed	1.61
PLA010-1	47	Failed	8.51
PLA016-1	1	3	3.00
PLA018-1	46	Failed	8.70
SET018-3	63	Failed	6.35
SET028-3	15	56	3.73
SET248-2	1	66	66.00
SET582+3	7	47	6.71
SET787-2	16	Failed	25.00
SET795+3	18	Failed	22.22
SET802+4	20	*41	2.05
SET811+4	24	275	11.46
SET812+4	17	73	4.29
SET831-2	92	*335	3.64
SET853-1	22	12	0.55
SET951+1	30	Failed	13.33
SEU003+1	78	302	3.87
SWV050+1	159	Failed	2.52
SWV282-2	3	191	63.67
SYN014-1	5	45	9.00
SYN036-3	52	23	0.44
		Average speedup	14.93

Table 7. Comparing Octopus with the learning strategy and Octopus without the learning strategy on the 31 theorems that Theo didn't prove.

9. Observations

It comes as no surprise that 133 computers using different proof strategies outperform a single computer. However, in general, as the number of computers in a multiprocessing system increases, it becomes increasingly difficult to find sufficiently different effective strategies. The learning strategy described in this paper creates an abundance of different ways to try to prove a given theorem. Most ways are of little help, but the best one can be very effective. Of course, determining the best one is the issue. For Octopus, the low-numbered slaves using the low-numbered weakenings on the Weakened Clause List proved the majority of the theorems. Nevertheless, the higher numbered slaves made a significant contribution to the performance of Octopus. This bodes well for Octopus running on yet larger systems.

The results for Theo show that using the first weakening on the Weakened Clause List is only marginally effective. For Octopus, using the proofs of weakened theorems in the way it does appears to be counterproductive most of the time. However, for the right weakened proof, the method is very effective. Again, finding the right one is the trick, and the more computers available for this effort, the better.

The weakenings created by Octopus are not all minimal. Moreover, many minimal weakening are not created. This is a consequence of the extra programming effort necessary to generate just the minimal weakenings. It seems though that this effort is called for at this point, and it should be at the top of the list of improvements to the program in the coming months.

One might argue that the learning strategy is a self-fulfilling prophecy: clauses assigned to the set-of-support are searched more deeply and more likely to be used in proofs. The degree to which this is the case is hard to measure as there are often many different ways to prove a theorem. One might carry out an experiment randomly assigning axioms to the set-of-support and compare their usage in proofs with the results of Section 8.3. However, from the tests on Theo, adding ineffective axioms to the set-of-support results in time wasted searching in the wrong direction, possibly negating the argument of a self-fulfilling prophecy. Overall, the results obtained by Octopus are quite impressive and encourage further experimentation with variations on the learning strategy described here.

Acknowledgements

Over the years, the systems staff at McGill's School of Computer Science has been supportive of this work and deserves the first thanks. This includes Andrew Bogecho, Ron Hansen, and the head of operations, Phil Ciaravola. In addition, some development during the last year was carried on the Barcelona Supercomputer Center's MareNostrum computer, currently the largest computer in Europe. The Center's support was greatly appreciated and useful. This effort is supported by the Canadian National Science and Engineering Research Council, and their financial support is acknowledged and thanked here as well.

References

[1] http://www.cs.miami.edu/~tptp/CASC/J3/

[2] http://www.cs.miami.edu/~tptp/index.html

[3] Automated Theorem Proving: Theory and Practice, Springer, 2001.

[4] Z. Wang and M. Newborn, "Octopus: Combining learning and parallel search," *Journal of Automated Reasoning*, v. 33, n. 2, pp. 171-218, 2004.

[5] M. Newborn and Z. Wang, Proofs of Nineteen More 1.0 Rated Theorems in the TPTP Problem Library, McGill University School of Computer Science Technical Report SOCS-04.1, 2004. (See http://www.cs.mcgill.ca/research/tech2003.html)

[6] M. Newborn, Proofs of Twenty-Four 1.0-Rated Theorems in the TPTP Problem Library, McGill

University School of Computer Science Technical Report SOCS-03.6, 2003. (See http://www.cs.mcgill.ca/research/tech2004.html)

[7] http://www.cs.miami.edu/~tptp/CASC/J3/WWWFiles/Results.html

[8] J. Meng and L. Paulson, Lightweight Relevance Filtering for Machine-Generated Resolution, ESCOR, 2006.

APPENDIX 1: CASC Systems Competition: Detailed Results of Top Eight Finishers in FOF and CNF Divisions with Data from Octopus Added.

Data for Octopus shows the time to find a proof exists, a slash, and then the time to complete the construction. Note that the CNF theorems on which Octopus failed because the theorem was too large have been moved to the end of the table.

FOF-2006	Vampire 8.1	E.99	EP.99	Vampire 8.0	Octopus Run 1	Octopus Run 2	Theo	Geo 2000i
AGT005+2	3.7	0.3	0.4	0.1	6/6	4/4	0.9	Timeout
AGT007+2	167.8	Timeout	Timeout	Timeout	**161/407**	**154/271**	Timeout	Timeout
AGT015+1	0	1.4	1.9	0	7/7	4/4	5.3	Timeout
AGT018+1	14.8	Timeout	Timeout	95.1	383/421	Fail	Timeout	Timeout
AGT024+2	29.9	Timeout	Timeout	Timeout	52/93	52/93	264.5	Timeout
ALG074+1	55.2	1.8	3.4	118.3	3/4	3/4	Timeout	1.3
ALG211+1	0	0	0.1	0	6/6	5/5	0	0
CSR001+1	357.3	75.5	87.1	Timeout	Fail	Fail	Timeout	Timeout
CSR002+1	95	26.8	28.2	Timeout	Fail	Fail	Timeout	Timeout
CSR005+2	76.5	34.3	35.7	Unknown	Fail	Fail	Timeout	Timeout
CSR007+1	1.1	46.8	51.6	2.2	153/175	110/128	Timeout	Timeout
CSR019+1	0	0.3	0.4	Unknown	5/6	5/5	1	Timeout
CSR021+1	0	0.2	0.4	Unknown	7/7	6/6	0.5	31.8
CSR022+1	0	0.4	0.6	Unknown	11/11	6/6	0.5	20.2
CSR024+1.009	0.6	0.4	0.7	Unknown	Fail!!!	Fail!!!	Unknown	Timeout
GEO085+1	0	0	0.1	0	6/6	6/6	0.2	0.7
GEO088+1	115.1	Timeout	Timeout	34.8	**4/5**	**4/5**	Timeout	Timeout
GEO109+1	120.6	Timeout	Timeout	32.1	Fail	Fail	Timeout	Timeout
GRA002+1*	Timeout	Timeout	Timeout	Unknown	Fail	Fail	Timeout	13.6
GRA002+2*	49.9	Timeout	Timeout	Unknown	Fail	Fail	Timeout	15.9
GRA002+4*	0	0	0.1	Unknown	5/5	5/5	0.2	0.9
GRA004+1*	0.1	0	0.1	Unknown	9/9	10/10	3.3	4.1
GRA006+1*	12	0.5	0.7	Unknown	Fail	Fail	Timeout	7
GRA007+1*	15	0.3	0.4	Unknown	218/443	302/354	Timeout	3
GRA007+2*	14.9	0.5	0.6	Unknown	264/535	270/550	Timeout	12.9
GRA008+1*	23.4	3.6	4.1	Unknown	Fail	Fail	Timeout	5.4
GRA009+2*	139.7	4.8	5.8	Unknown	Fail	Fail	Timeout	4.3
GRA010+1*	0	0.1	0.1	Unknown	1/1	1/1	0.9	1.5
GRA011+1*	Timeout	Timeout	Timeout	Unknown	Fail	Fail	Timeout	15
GRA012+1*	Timeout	Timeout	Timeout	Unknown	Fail	Fail	Timeout	13.3
GRP001+6	0.3	0.4	0.7	1.8	2/3	5/5	0.6	0.2
GRP012+5	3.6	0.1	0.2	19.8	3/3	3/3	0.1	10.2
LCL414+1	0	0	0.1	0	8/8	7/7	0	0.6
MGT022+1	0	0	0.1	0	5/5	9/9	0	0.1
MGT022+2	0	0	0.1	0	4/4	4/4	0	0.1
MGT028+1	0	0	0.1	0	6/6	5/5	0	0
MGT036+1	0	0	0.1	0	8/8	6/6	0	1
MGT036+3	0	0	0.1	0	6/6	6/6	0	0
MGT042+1	22.9	Timeout	Timeout	84.4	12/12	12/12	131	3.2

FOF-2006	Vampire 8.1	E.99	EP.99	Vampire 8.0	Octopus Run 1	Octopus Run 2	Theo	Geo 2000i
MGT060+1	8.9	0	0.1	0.6	6/6	2/2	0.2	0.4
MSC010+1	2.7	GaveUp	GaveUp	0	Fail!!!	Fail!!!	Unknown	Timeout
NUM310+1	11.7	Timeout	Timeout	79.9	12/20	13/30	19	Timeout
NUM327+1	11	Timeout	Timeout	79.7	7/7	7/7	25.7	Timeout
NUM329+1	10.8	Timeout	Timeout	83.6	7/8	8/8	26.3	Timeout
NUM334+1	10.7	Timeout	Timeout	79.6	7/9	7/8	5.9	Timeout
NUM398+1*	0	0.1	0.2	0	2/2	2/2	4	0.2
NUM400+1*	81.1	0.1	0.2	0.7	30/47	30/47	53.1	149.7
PUZ047+1	0	0	0.1	0	7/7	8/8	0	0.1
SET008+3	0	0.1	0.2	0	3/3	6/6	0.1	63
SET009+3	0	0	0.1	0	4/4	6/6	0.5	43
SET014+3	0.1	0	0.2	0.1	7/7	6/6	0.1	1.8
SET067+1	0	0.1	0.2	0.1	5/6	5/5	0.6	Timeout
SET094+1	0	0	0.1	0	1/2	1/2	1.5	Timeout
SET099+1	Timeout	57.4	62.9	182.3	Fail	Fail	Timeout	Timeout
SET144+3	173	Timeout	Timeout	103.6	6/7	4/8	15.9	285.8
SET171+3	12.2	Timeout	Timeout	102.6	6/21	5/20	42	Timeout
SET200+3	2	0.1	0.2	0.2	7/7	6/6	0.1	12.4
SET577+3	0.1	0.2	0.5	0.2	3/3	3/4	0.1	5.4
SET579+3	0.1	0	0.1	0.1	1/1	1/1	6.3	48.1
SET582+3	145.9	36.9	49.9	115.6	**7/21**	**7/21**	Timeout	Timeout
SET584+3	1.5	0	0.2	0	5/5	6/6	0	7.1
SET587+3	0.1	0	0.1	0	7/7	4/6	7.2	7.6
SET588+3	0	0	0.1	0	7/7	8/8	0.4	18
SET589+3	0.2	0	0.1	0	6/6	5/5	0	0.3
SET590+3	0	0	0.1	0	4/4	5/5	0	0
SET593+3	0.1	0.1	0.2	0.6	15/15	8/8	7.7	60.9
SET594+3	0.6	1.1	2.4	0	6/6	10/10	0.7	100.1
SET595+3	22.1	26.6	39.7	16.8	3/3	2/2	1.4	318.5
SET603+3	0	0	0.1	0	6/6	6/7	0	0.2
SET623+3	Timeout	Timeout	Timeout	Timeout	Fail	Fail	Timeout	Timeout
SET624+3	4.9	0	0.1	1.4	6/6	3/3	1.3	232.6
SET631+3	0	0	0.1	0	6/6	6/6	0	0.5
SET655+3	1.5	0.4	0.6	204.1	2/3	1/1	6.9	Timeout
SET656+3	3.9	38.4	49.8	12.7	3/3	4/4	0.1	Timeout
SET657+3	140.9	0.1	0.2	16.8	1/2	2/2	0.7	Timeout
SET669+3	6.7	0.9	1.3	0.2	4/4	6/7	3.8	Timeout
SET680+3	0.1	0	0.1	0.5	8/8	9/10	1.9	Timeout
SET795+4*	18	Timeout	Timeout	0.1	19/22	18/22	Timeout	Timeout
SET796+4*	Timeout	Timeout	Timeout	Timeout	Fail	Fail	Timeout	Timeout
SET798+4*	11.9	0.1	0.2	39.6	5/5	7/7	0.2	Timeout
SET802+4*	109.5	4.1	5.3	25.8	27/27	20/20	Timeout	Timeout
SET805+4*	Timeout	Timeout	Timeout	343.6	Fail	Fail	Timeout	Timeout
SET806+4*	Timeout	Timeout	Timeout	208.6	Fail	Fail	Timeout	Timeout
SET808+4*	Timeout	Timeout	Timeout	Timeout	Fail	Fail	Timeout	Timeout
SET811+4*	Timeout	Timeout	Timeout	207.1	17/32	24/46	Timeout	Timeout
SET812+4*	Timeout	Timeout	Timeout	104.1	19/29	17/38	Timeout	Timeout
SET814+4*	Timeout	2.9	4.1	10.9	2/2	1/1	0.2	Timeout

FOF-2006	Vampire 8.1	E.99	EP.99	Vampire 8.0	Octopus Run 1	Octopus Run 2	Theo	Geo 2000i
SET815+4*	Timeout	Timeout	Timeout	Timeout	Fail	Fail	GaveUp	Timeout
SET816+4*	Timeout	Timeout	Timeout	Timeout	Fail	Fail	Timeout	Timeout
SET896+1*	0	0	0.1	0	6/6	6/6	0	0
SET899+1*	0	0	0.1	0	6/6	6/6	0	0
SET913+1*	0	0	0.1	0	5/5	5/5	0	0
SET915+1*	0	0	0.1	0	7/7	5/5	0	0
SET919+1*	Timeout	Timeout	Timeout	Timeout	**3/6**	**3/6**	21.2	Timeout
SET935+1*	0.1	0.1	0.3	0.2	2/2	2/3	1.9	Timeout
SET939+1*	0	0	0.1	0	6/6	6/6	0	0
SET942+1*	0	0	0.1	5.9	4/4	5/5	0.1	6.3
SET951+1*	0.1	Timeout	Timeout	0	30/41	30/41	Timeout	Timeout
SET966+1*	0	0	0.1	0	4/4	6/6	0.1	0.1
SET982+1*	0.1	95.5	99.9	18.7	Fail	Fail	Timeout	Timeout
SEU003+1*	35.7	145.6	164.2	149.9	84/84	78/78	GaveUp	Timeout
SEU009+1*	0.2	0	0.1	0	8/11	10/10	5.9	69.1
SEU014+1*	Timeout	Timeout	Timeout	204.5	Fail	Fail	Timeout	26.1
SEU075+1*	Timeout	Timeout	Timeout	106.8	Fail--Bug	Fail	GaveUp	Timeout
SEU086+1*	Timeout	Timeout	Timeout	Timeout	Fail	Fail	Timeout	16.5
SWC078+1	Timeout	95.7	118.2	Timeout	Fail	Fail	Timeout	Timeout
SWC231+1	21.5	2.2	2.5	60	Fail	Fail	Timeout	Timeout
SWC239+1	11.2	0.1	0.2	1.3	44/44	39/39	159.2	Timeout
SWV011+1	0	0	0.1	0	5/5	5/5	0	20.5
SWV014+1	0	0	0.1	0.1	1/1	1/1	0.7	Timeout
SWV026+1	22	Timeout	Timeout	Timeout	Fail	Fail	Timeout	Timeout
SWV030+1	20.1	0.1	0.2	Timeout	60/60	15/15	376.6	Timeout
SWV033+1	54	5.4	8.7	Timeout	Fail	Fail	Timeout	Timeout
SWV036+1	78.6	Timeout	Timeout	Timeout	Fail!!!	Fail!!!	Unknown	Timeout
SWV038+1	22.2	Timeout	Timeout	Timeout	Fail!!!	Fail!!!	Timeout	Timeout
SWV039+1	21.9	Timeout	Timeout	Timeout	Fail	Fail	Timeout	Timeout
SWV041+1	0	0.1	0.2	0.1	1/1	2/3	2	Timeout
SWV048+1	4.9	0.1	0.2	0.2	6/6	6/7	124.5	Timeout
SWV050+1	4.9	0	0.1	0.2	236/246	159/159	Timeout	Timeout
SWV053+1	62.5	13.4	17.9	Timeout	Fail	Fail	Timeout	Timeout
SWV090+1	Timeout	2	3.6	Timeout	Fail	Fail	Timeout	Timeout
SWV092+1	5.3	0.1	0.2	0.2	2/2	2/2	2.6	Timeout
SWV094+1	21	12.7	15.9	Timeout	Fail	Fail	Timeout	Timeout
SWV096+1	6.6	2	4.5	12.2	2/2	2/2	6.7	Timeout
SWV098+1	6.5	0.1	0.2	0.5	8/8	8/8	64.2	Timeout
SWV103+1	Timeout	2.4	3.9	Timeout	Fail	Fail	Timeout	Timeout
SWV117+1	15.4	Timeout	Timeout	Timeout	Fail	Fail	Timeout	Timeout
SWV161+1	17.3	0.1	0.2	129.6	3/3	4/5	3.4	Timeout
SWV173+1	5.2	0	0.1	0.1	2/2	1/1	1.2	1.3
SWV177+1	5.2	0	0.1	0	2/2	2/2	2.9	Timeout
SWV200+1	6	0.4	0.9	0.3	6/6	6/6	1.5	Timeout
SWV209+1	52.7	0.1	0.2	162.7	1/1	2/3	4.3	Timeout
SWV233+1*	130.1	0.2	0.3	Timeout	Fail	Fail	Timeout	Timeout
SWV234+1*	Timeout	Timeout	Timeout	255.6	Fail	Fail	Timeout	Timeout
SWV234+2*	0.8	0	0.2	49.4	Fail	Fail	Timeout	Timeout

FOF-2006	Vampire 8.1	E.99	EP.99	Vampire 8.0	Octopus Run 1	Octopus Run 2	Theo	Geo 2000i
SWV235+1*	0.2	0.2	0.3	4.2	1/1	1/1	1.6	Timeout
SWV236+1*	0.1	0	0.1	0.9	1/1	2/3	2	Timeout
SWV237+1*	0	0	0.1	12.4	5/5	5/5	0.9	3.8
SYN081+1	0	0	0.1	0	6/6	6/6	0.1	0
SYN084+1	0	0	0.1	0	1/1	1/2	0.1	0
SYN364+1	0	0	0.1	0	5/5	6/6	0	0
SYN365+1	0	0	0.1	0	3/3	7/7	0	0
SYN728+1	0	0	0.1	0	5/5	5/5	0	0
SYN730+1	0	0	0.1	0	6/6	7/7	0.1	0
SYN938+1	0	2.1	3	0.1	Fail!!!	Fail!!!	Unknown	2.6
SYN939+1	0	0	0.1	0	8/8	8/8	0.1	0
SYN940+1	0	0	0.1	0	9/9	8/8	0	0
SYN941+1	0	0	0.1	0	7/7	7/7	0	0
SYN943+1	0	0	0.1	0	4/4	5/5	0	0
SYN980+1	0	0	0.1	0	4/4	7/7	0	0

FOF-2006	Vampire 8.1	E.99	EP.99	Vampire 8.0	Octopus Run 1	Octopus Run 2	Theo	Geo 2000i
Attempted	150	150	150	150	150	150	150	150
Solved	128	112	112	109	109	108	94	73
Solutions	128	0	92	109	105	107	94	73
New Attempted	48	48	48	48	48	48	48	48
New Solved	31	29	29	29	28	28	20	25

CNF-2006	Vampire 8.1	Vampire 8.0	E.99	EP.99	Octopus	Theo	Otter 3.3	Darwin 1.3
ANA002-1	3.3	2.3	19.2	29.8	3/7	+231.90	Timeout	Timeout
ANA003-2	161.8	23.2	10.1	13.1	4/10	+228.30	Timeout	Timeout
ANA004-5	7	Timeout	Timeout	Timeout	4/4	+279.80	Timeout	Timeout
ANA008-2*	0	0	0	0.1	7/7	+0.00	0	0
ANA014-2*	0	0	0	0.1	Fail	Timeout	12.7	Timeout
ANA016-2*	0	0	0	0.1	1/1	+9.40	0	Timeout
ANA017-2*	0	0	0	0.1	3/3	+20.40	0	4.3
ANA030-2*	0.2	0	0	0.1	Fail	Timeout	0.1	Timeout
ANA040-2*	0	0	0	0.1	3/3	+0.00	0	0
ANA044-2*	0	0	0	0.1	5/5	+0.20	0	0.1
ANA045-2*	0	0	0	0.1	4/4	+0.10	0.1	0
BOO020-1	11.4	1.2	9.9	19.8	Fail	Timeout	Timeout	Timeout
COL003-10	40.5	11.4	25.2	29.7	Fail	Timeout	Unknown	Timeout
COL003-2	0.8	1.4	0.4	0.8	51/93	Timeout	Unknown	Timeout
COL042-3	39	6.9	5.7	9.9	Fail	Timeout	GaveUp	Timeout
COL093-2	Timeout	Timeout	0.2	0.4	114/114	Timeout	Timeout	Timeout
COL112-1*	0.1	0.1	0.8	1.1	5/5	+2.80	2.6	0.3
COL119-1*	0.1	0.1	0.9	1.1	5/5	+2.90	4.3	0.3
COM005-1	183.6	6.4	19.7	29.8	Fail	Timeout	Timeout	Timeout
COM009-1*	0.1	0.1	0.9	1.1	23/23	+64.70	4.5	Timeout
FLD013-1	2	2.5	22.8	29.8	4/4	+1.20	Timeout	2.7
FLD060-3	0.2	1	Timeout	Timeout	1/1	+1.10	Timeout	29.5

NF-2006	Vampire 8.1	Vampire 8.0	E.99	EP.99	Octopus	Theo	Otter 3.3	Darwin 1.3
GEO006-1	1.2	0.2	0	0.1	4/4	+14.30	271.2	Timeout
GEO007-1	1.3	58.8	152.1	179.5	11/19	+308.70	Timeout	Timeout
GEO011-1	1.1	0.1	0.5	1.1	2/4	+83.50	Timeout	Timeout
GEO051-3	1.2	14.7	0.5	0.9	1/1	+2.90	7.8	Timeout
GEO076-4	0.1	11.9	0	0.1	8/12	Timeout	Timeout	0.2
GEO083-1	0	7.5	25.8	29.2	1/1	+0.50	Timeout	Timeout
GEO089-1	66.4	1.8	Timeout	Timeout	36/37	Timeout	Timeout	Timeout
GRP039-5	0.1	0.1	0	0.1	18/45	+274.70	Timeout	Timeout
GRP061-1	15.1	13.2	0.6	1.3	Fail	Timeout	GaveUp	Timeout
GRP070-1	11.2	0.2	0	0.1	Fail	Timeout	1	Timeout
GRP074-1	0.1	3.9	0.1	0.3	Fail	Timeout	GaveUp	Timeout
GRP082-1	0.2	11.1	0.1	0.3	Fail	Timeout	GaveUp	Timeout
GRP085-1	0	0.1	0.1	0.4	Fail	Timeout	0.1	1.1
GRP086-1	0	0	0.1	0.2	Fail	Timeout	0.1	150.8
GRP102-1	1.6	0.2	0	0.1	Fail	Timeout	2.5	Timeout
GRP103-1	2.6	0.5	0.9	2.9	Fail	Timeout	8.4	Timeout
GRP110-1	0	0.1	0.1	0.5	Fail	Timeout	17.7	109.4
GRP311-1	1.1	2.9	2.1	2.7	249/249	Timeout	Timeout	Timeout
HWV002-1	26	42.6	37.3	49.6	Fail	Timeout	GaveUp	Timeout
HWV003-2	4.2	40	86.7	109.4	Fail	Timeout	GaveUp	Timeout
HWV016-1	0	0.2	0.1	0.2	4/7	+109.20	Timeout	Timeout
LAT003-1	Timeout	Timeout	72.8	89.5	Fail	Timeout	Timeout	Timeout
LAT005-4	21.8	12.2	1.2	3.5	Fail	Timeout	140.5	Timeout
LAT198-1	55.5	332	3.8	9.4	Fail	Timeout	Timeout	Timeout
LAT201-1	Timeout	Timeout	7	9.9	Fail	Timeout	Timeout	Timeout
LAT209-1	22.6	31.4	3.3	8.6	Fail	Timeout	GaveUp	Timeout
LAT219-1	Timeout	Timeout	Timeout	Timeout	Fail	Timeout	Timeout	Timeout
LAT263-2*	0	0	0	0.1	3/3	+0.80	0	0.1
LAT273-2*	0	0	0	0.1	6/6	+0.00	0	0
LAT277-2*	0	0	0	0.1	5/5	+0.00	0	0
LCL028-1	1	197.3	Timeout	Timeout	Fail	Timeout	Timeout	Unknown
LCL109-4	25.8	33.5	6	9.8	Fail	Timeout	Timeout	Timeout
LCL124-1	3.4	23.2	0.2	0.4	Fail	Timeout	Timeout	Timeout
LCL125-1	99.7	68.1	Timeout	Timeout	Fail	Timeout	161.3	Timeout
LCL145-1	0	0	0	0.1	Fail	GaveUp	Timeout	Timeout
LCL151-1	169.2	89.5	Timeout	Timeout	Fail	Timeout	Timeout	Timeout
LCL185-3	0	0	2	3.3	3/3	+0.20	0.3	0.1
LCL222-3	30.1	21.7	52	68.7	Fail	Timeout	GaveUp	Timeout
LCL230-1	3.9	0.4	3	4.1	1/1	+2.10	GaveUp	Timeout
LCL237-3	0.1	0	0.4	0.8	6/6	+86.10	1.5	79.8
LCL249-1	1.7	14.6	Timeout	Timeout	Fail	Timeout	GaveUp	Timeout
LCL253-1	27.5	125.1	Timeout	Timeout	Fail	Timeout	GaveUp	Timeout
LCL423-1	Timeout	Timeout	Timeout	Timeout	Fail	Timeout	GaveUp	Timeout
LCL434-2*	0	0	0	0.1	4/4	+0.10	0	0
LCL439-1*	26.9	23.8	Timeout	Timeout	30/30	+18.00	Timeout	Timeout
LCL442-2*	0.1	0	0	0.1	7/7	+0.30	14.6	0
LCL444-2*	80	47.4	Timeout	Timeout	Fail	GaveUp	GaveUp	Timeout
MGT034-2	0.2	0	0.4	0.5	1/1	+2.00	81.9	Timeout
NLP080-1	0	0	0	0.1	Fail!!	GaveUp	0	0

NF-2006	Vampire 8.1	Vampire 8.0	E.99	EP.99	Octopus	Theo	Otter 3.3	Darwin 1.3
NUM010-1	74.4	9.7	Timeout	Timeout	18/37	+125.30	Timeout	Timeout
NUM017-1	76.2	21.4	10.9	14.2	1/1	+88.30	47.2	Timeout
NUM058-1	77.5	92.8	Timeout	Timeout	Fail	Timeout	Timeout	Timeout
PLA010-1	0.2	0.1	0.1	0.2	47/47	Timeout	Timeout	Timeout
PLA014-1	0.1	0	0	0.1	2/2	+58.50	Timeout	Timeout
PLA016-1	0.1	0	0	0.1	1/1	Timeout	Timeout	0.5
PLA018-1	0.1	0	0	0.1	46/75	Timeout	Timeout	Timeout
PUZ035-4	0	0	0	0.1	8/8	+4.50	0	0
RNG004-3	175.4	144.5	6.8	9.9	1/1	+2.20	34.2	Timeout
RNG027-2	Timeout	75.1	Timeout	Timeout	Fail	Timeout	Timeout	Timeout
RNG029-3	0.7	2	6.7	9.9	Fail	Timeout	Timeout	Timeout
SET013-1	1.2	0	0.6	0.9	3/3	+125.50	Timeout	47.7
SET016-3	0	47.8	0.2	0.4	Fail	Timeout	Timeout	Timeout
SET018-3	0	4.9	126.6	156.6	63/95	Timeout	Timeout	Timeout
SET018-4	0.1	56.1	118.6	139.4	Fail	Timeout	Timeout	Timeout
SET022-3	0.8	0.4	8.9	13	2/2	+1.30	Timeout	Timeout
SET027-3	0	0.1	0.1	0.2	7/7	+0.60	224	318.6
SET028-3	35.6	124.9	Timeout	Timeout	15/22	Timeout	Timeout	Timeout
SET035-3	117.9	113.8	Timeout	Timeout	Fail	Timeout	Timeout	Timeout
SET208-6	3.4	18	Timeout	Timeout	19/23	+214.10	Timeout	Timeout
SET787-2	Unknown	101.4	0.1	0.2	16/16	Timeout	Timeout	Timeout
SET820-2*	0	0	0	0.1	7/7	+0.10	0	0
SET823-1*	0.1	0.1	2.1	2.6	3/3	+3.00	Timeout	Timeout
SET827-1*	0.1	0.1	1	1.2	5/5	+1.20	2.5	0.2
SET830-2*	0.1	5.8	0	0.1	1/1	+0.40	Timeout	2.8
SET831-2*	0.4	74.5	0.1	0.2	92/92	Timeout	0.4	Timeout
SET834-1*	62.1	243	9.1	10.5	Fail	Timeout	Timeout	Timeout
SET834-2*	0	3.2	0	0.1	1/1	+0.70	GaveUp	2.8
SET838-2*	0	0	0	0.1	3/4	+0.10	0	0
SET841-1*	0.1	5.2	0.9	1.1	5/7	+159.80	3.3	16.6
SET853-1*	0.3	75.1	Timeout	Timeout	22/22	Timeout	Timeout	Timeout
SET861-2*	12.6	12.3	Timeout	Timeout	Fail	Timeout	GaveUp	Timeout
SET863-2*	0	0.2	0	0.1	1/1	+0.10	4.2	0.1
SWC169-1	Timeout	Timeout	299	307.4	Fail	Timeout	Timeout	Timeout
SWC225-1	49.6	22.8	0.1	0.2	Fail	Timeout	Timeout	Timeout
SWV014-1	6.7	0.1	0	0.1	1/1	+0.90	Timeout	Timeout
SWV239-1*	14.9	88.6	2.3	2.7	Fail	Timeout	Timeout	Timeout
SWV242-1*	1.2	0.8	3.5	4.1	11/12	+262.50	Timeout	Timeout
SWV247-1*	52.6	162	Timeout	Timeout	Fail	Timeout	Timeout	Timeout
SWV248-2*	1.2	0	0	0.1	1/1	Timeout	GaveUp	0.1
SWV249-2*	4	18.9	Timeout	Timeout	Fail	Timeout	1.2	Timeout
SWV251-2*	0	0	0.1	0.3	7/8	+8.70	0.8	Timeout
SWV264-1*	0.1	0.1	0.6	0.7	5/5	+6.20	3.8	0.2
SWV274-2*	0	0	0	0.1	3/4	+0.20	7.9	0
SWV275-2*	1.2	2.9	2.3	3.6	2/2	+0.30	Timeout	0
SWV281-2*	0	0	0	0.1	5/5	+0.10	Timeout	Timeout
SWV282-2*	0	0.2	0.6	1.1	3/3	Timeout	Timeout	Timeout
SWV291-2*	0	0	0	0.1	7/7	+0.10	0	0
SWV302-1*	1	0.4	1	1.2	5/5	+14.20	GaveUp	Timeout

NF-2006	Vampire 8.1	Vampire 8.0	E.99	EP.99	Octopus	Theo	Otter 3.3	Darwin 1.3
SWV305-2*	0	0	0	0.1	6/7	+0.20	0	Timeout
SWV319-2*	0	0	0	0.1	1/2	+0.50	0	Timeout
SWV323-2*	0.8	0	0	0.1	4/5	+0.50	6.6	0.1
SWV325-2*	0	0	0	0.1	7/7	+0.60	0	2
SWV329-2*	0	0	0	0.1	6/6	+0.00	0	0
SWV354-2*	0	0	0	0.1	Fail	Timeout	9.3	Timeout
SYN014-1	0	0	0	0.1	5/5	Timeout	Timeout	0
SYN015-1	0	0.1	0	0.1	Fail	Timeout	Timeout	0
SYN036-3	0	0	0	0.1	52/52	Timeout	0.1	0
SYN067-3	0.8	8.1	4	6.1	Fail	Timeout	Timeout	0.1
SYN600-1	25.2	5.4	6.5	9.6	1/1	+0.50	4.9	Timeout
SYN640-1	2	0.6	Timeout	Timeout	5/5	+9.90	1.6	Timeout
SYN707-1	25.6	3.6	0.1	0.1	6/6	+2.20	Timeout	Timeout
SYN708-1	25.6	3.5	1.5	1.9	1/1	+2.80	Timeout	Timeout
ANA022-1*	211.1	Timeout	Timeout	Timeout	Fail!!!	Unknown	GaveUp	Timeout
ANA037-1*	6.7	41.2	180.1	182.9	Fail!!!	Unknown	GaveUp	Timeout
SWV243-1*	1.1	1.6	6.2	6.7	Fail!!!	Unknown	GaveUp	Timeout
SWV254-1*	1.4	2.4	6.9	7.4	Fail!!!	Unknown	GaveUp	Timeout
SWV256-1*	0.8	0.9	0.6	0.8	Fail!!!	Unknown	GaveUp	3.6
SWV267-1*	1.4	15.9	7.2	7.8	Fail!!!	Unknown	GaveUp	3.5
SWV270-1*	1.6	36	6.3	6.6	Fail!!!	Unknown	GaveUp	Timeout
SWV273-1*	1.3	4	6.3	6.9	Fail!!!	Unknown	GaveUp	Timeout
SWV315-1*	2	36.3	7	7.5	Fail!!!	Unknown	GaveUp	Timeout
SWV317-1*	1.4	3	6.7	7.3	Fail!!!	Unknown	GaveUp	Timeout
SWV341-1*	1.4	18.6	8	8.5	Fail!!!	Unknown	GaveUp	Timeout
SWV342-1*	84.7	Timeout	11.4	12.4	Fail!!!	Unknown	GaveUp	Timeout
SWV343-1*	82.8	Timeout	12.5	13.1	Fail!!!	Unknown	GaveUp	Timeout
SWV349-1*	0.2	0.3	0.2	0.4	Fail!!!	Unknown	0.9	4.9
SWV354-1*	Timeout	Timeout	393	Timeout	Fail!!!	Unknown	GaveUp	Timeout
SWV355-1*	83.6	Timeout	8.9	9.5	Fail!!!	Unknown	GaveUp	Timeout
NF-2006	**Vampire 8.1**	**Vampire 8.0**	**E.99**	**EP.99**	**Octopus**	**Theo**	**Otter 3.3**	**Darwin 1.3**
Attempted	150	150	150	150	150	150	150	150
Solved	141	138	126	125	83	66	55	47
Solutions	141	138	0	103	83	66	55	0
New Attempted	65	65	65	65	65	65	65	64
New Solved	64	60	58	57	40	36	34	30

APPENDIX 2: Octopus's Proof of SET919+1

```
----------------------------------------------------------------------
                         MODIFY THEOREM

Time taken = 0   Time to go = 400   Maxtime on this try = 200

Weakened to: 0 43S ~E.[f3[x,SK7],f4[SK6]] ... ?

--------------------------------------------------------------------
   0 <BC: 44 NC: 3 AC: 45 U: 64>
1 {T0 N782-782 [C2101 H22 h0 U26656 u51634 f0] E3 s332 t0 L0}
2 {T1 N7953-7953 [C22676 H2320 h0 U219764 u412636 f0] E3 s2449 t0 L0}
3 {T2 N16653-16653 [C54504 H7412 h0 U241381 u505258 f0] E15 s5640 t0 L0}*

Weakened proof obtained; details soon!!

1 {T2 N17435-782 [C2101 H22 h0 U26656 u51634 f0] E3 s5972 t0 L0}
2 {T4 N24606-7953 [C22676 H2320 h0 U219764 u412636 f0] E3 s8089 t0 L0}
3 {T5 N32985-16332 [C53405 H7240 h0 U241346 u504755 f0] E12 s11116 t0 L0}$

THE W-PROOF


Axioms:
 1: E.[x,f3[y,z]] ~p1[SK2[y,z,x],x] ~p1[SK2[y,z,x],y] ~p1[SK2[y,z,x],z]
 2: E.[x,f2[y,z]] E.[SK1[y,z,x],y] E.[SK1[y,z,x],z] p1[SK1[y,z,x],x]
 3 >~E.[x,f3[y,z]] p1[u,x] ~p1[u,y] ~p1[u,z]
 4: ~E.[x,f2[y,z]] ~p1[u,x] E.[u,y] E.[u,z]
 5: E.[x,f3[y,z]] p1[SK2[y,z,x],z] p1[SK2[y,z,x],x]
 6: E.[x,f3[y,z]] p1[SK2[y,z,x],y] p1[SK2[y,z,x],x]
 7: E.[x,f2[y,z]] ~p1[SK1[y,z,x],x] ~E.[SK1[y,z,x],z]
 8: E.[x,f2[y,z]] ~p1[SK1[y,z,x],x] ~E.[SK1[y,z,x],y]
 9 >E.[x,f4[y]] E.[SK3[y,x],y] p1[SK3[y,x],x]
10 >E.[x,f4[y]] ~p1[SK3[y,x],x] ~E.[SK3[y,x],y]
11 >~E.[x,f3[y,z]] ~p1[u,x] p1[u,z]
12: ~E.[x,f3[y,z]] ~p1[u,x] p1[u,y]
13: ~E.[x,f2[y,z]] p1[u,x] ~E.[u,z]
14: ~E.[x,f2[y,z]] p1[u,x] ~E.[u,y]
15 >~E.[x,f4[y]] p1[z,x] ~E.[z,y]
16 >~E.[x,f4[y]] ~p1[z,x] E.[z,y]
17: ~E.[x,y] ~E.[y,z] E.[x,z]
18: ~p1[x,y] ~E.[y,z] p1[x,z]
19: ~p1[x,y] ~E.[x,z] p1[z,y]
20: ~p5[x] ~E.[x,y] p5[y]
21: ~E.[x,y] E.[SK1[x,z,u],SK1[y,z,u]]
22: ~E.[x,y] E.[SK1[z,x,u],SK1[z,y,u]]
23: ~E.[x,y] E.[SK1[z,u,x],SK1[z,u,y]]
24: ~E.[x,y] E.[SK2[x,z,u],SK2[y,z,u]]
25: ~E.[x,y] E.[SK2[z,x,u],SK2[z,y,u]]
26: ~E.[x,y] E.[SK2[z,u,x],SK2[z,u,y]]
27: ~E.[x,y] E.[f3[z,x],f3[z,y]]
28: ~E.[x,y] E.[f3[x,z],f3[y,z]]
29: ~E.[x,y] E.[f2[z,x],f2[z,y]]
30: ~E.[x,y] E.[f2[x,z],f2[y,z]]
31: ~E.[x,y] E.[SK3[x,z],SK3[y,z]]
32: ~E.[x,y] E.[SK3[z,x],SK3[z,y]]
33: ~E.[x,y] E.[f4[x],f4[y]]
34: ~p1[x,y] ~p1[y,x]
35: ~E.[x,y] E.[y,x]
36 >E.[f3[x,y],f3[y,x]]
37: E.[f2[x,y],f2[y,x]]
38 >E.[f3[x,x],x]
39 >E.[x,x]
40: p5[SK5]
41: ~p5[SK4]

Negated conclusion:
42S ~p1[SK8,SK7] E.[SK6,SK8]
43S>~E.[f3[x,SK7],f4[SK6]] ... ?
44S>p1[SK6,SK7]
```

```
---------------

Phases 1 and 2 clauses used in proof:
51S>(43a,9a)  E.[SK3[SK6,f3[x,SK7]],SK6] p1[SK3[SK6,f3[x,SK7]],f3[x,SK7]] ... ?
 52S>(51b,11b) E.[SK3[SK6,f3[x,SK7]],SK6] ~E.[f3[x,SK7],f3[y,z]] p1[SK3[SK6,f3[x,SK7]],z]
... ?
  53S>(52b,36a) E.[SK3[SK6,f3[x,SK7]],SK6] p1[SK3[SK6,f3[x,SK7]],x] ... ?
   54S>(53b,16b) E.[SK3[SK6,f3[x,SK7]],SK6] ~E.[x,f4[y]] E.[SK3[SK6,f3[x,SK7]],y] ... ?
    55S>(54b,38a) E.[SK3[SK6,f3[f3[f4[x],f4[x]],SK7]],SK6]
E.[SK3[SK6,f3[f3[f4[x],f4[x]],SK7]],x] ... ?
     56S>(55ab)    E.[SK3[SK6,f3[f3[f4[SK6],f4[SK6]],SK7]],SK6] ... ?

57S>(44a,3d)  ~E.[x,f3[y,SK7]] p1[SK6,x] ~p1[SK6,y]
 58S>(57a,39a) p1[SK6,f3[x,SK7]] ~p1[SK6,x]
  59S>(58b,15b) p1[SK6,f3[x,SK7]] ~E.[x,f4[y]] ~E.[SK6,y]
   60S>(59b,38a) p1[SK6,f3[f3[f4[x],f4[x]],SK7]] ~E.[SK6,x]
    61S>(60b,39a) p1[SK6,f3[f3[f4[SK6],f4[SK6]],SK7]]

62S>(43a,10a) ~p1[SK3[SK6,f3[x,SK7]],f3[x,SK7]] ~E.[SK3[SK6,f3[x,SK7]],SK6] ... ?
 63S>(62b,16c) ~p1[SK3[SK6,f3[x,SK7]],f3[x,SK7]] ~E.[y,f4[SK6]] ~p1[SK3[SK6,f3[x,SK7]],y]
... ?
  64S>(63b,38a) ~p1[SK3[SK6,f3[x,SK7]],f3[x,SK7]]
~p1[SK3[SK6,f3[x,SK7]],f3[f4[SK6],f4[SK6]]] ... ?
   65S>(64b,11c) ~p1[SK3[SK6,f3[x,SK7]],f3[x,SK7]] ~E.[y,f3[z,f3[f4[SK6],f4[SK6]]]]
~p1[SK3[SK6,f3[x,SK7]],y] ... ?
    66S>(65b,36a) ~p1[SK3[SK6,f3[x,SK7]],f3[x,SK7]]
~p1[SK3[SK6,f3[x,SK7]],f3[f3[f4[SK6],f4[SK6]],y]] ... ?
     67S>(66ab)    ~p1[SK3[SK6,f3[f3[f4[SK6],f4[SK6]],SK7]],f3[f3[f4[SK6],f4[SK6]],SK7]]
... ?
      68S>[67a,56a] ~p1[SK6,f3[f3[f4[SK6],f4[SK6]],SK7]] ... ?
       69S>(68a,61a) [] ... ?


END OF W-PROOF

List of useful clauses from weakened proof

53: (52b,36a) E.[SK3[SK6,f3[f2[SK6,SK8],SK7]],SK6]
p1[SK3[SK6,f3[f2[SK6,SK8],SK7]],f2[SK6,SK8]]
54: (53b,16b) E.[SK3[SK6,f3[f2[SK6,SK8],SK7]],SK6] ~E.[f2[SK6,SK8],f4[x]]
E.[SK3[SK6,f3[f2[SK6,SK8],SK7]],x]
60S (59b,38a) p1[SK6,f3[f3[f4[x],f4[x]],SK7]] ~E.[SK6,x]
61S (60b,39a) p1[SK6,f3[f3[f4[SK6],f4[SK6]],SK7]]
~p1[SK3[SK6,f3[f2[SK6,SK8],SK7]],f3[f4[SK6],f4[SK6]]]
65: (64b,11c) ~p1[SK3[SK6,f3[f2[SK6,SK8],SK7]],f3[f2[SK6,SK8],SK7]]
~E.[x,f3[y,f3[f4[SK6],f4[SK6]]]] ~p1[SK3[SK6,f3[f2[SK6,SK8],SK7]],x]
66: (65b,36a) ~p1[SK3[SK6,f3[f2[SK6,SK8],SK7]],f3[f2[SK6,SK8],SK7]]
~p1[SK3[SK6,f3[f2[SK6,SK8],SK7]],f3[f3[f4[SK6],f4[SK6]],x]]

End of list of useful clauses


---------------------------------------------------*/
ANALYSIS OF WEAKENED PROOF

Clause 43 was weakened:  43S>~E.[f3[f2[SK6,SK8],SK7],f4[SK6]]
Length of weakened proof is 19 inferences.
Nodes searched for weakened proof is 32985.
15 clauses in w-proof usable to prove theorem.
Of these, 6 will be added to the base.
9 of given 41 axioms used in w-proof placed in sos.
2 of given 3 sos clauses used in w-proof.

Proof of w-version may be useful.

Weakened clause used in proof.

-----------------------------------------------------*/

PHASE 0: 0 s, PHASE 1: 2 s, PHASE 2: 3 s ACCUM: 0  Total Time: 5 s
NOD: 32985            RES: 117359           FAC: 5604             T: 16  V: 8  L: 3
```

```
CTE: 54504            CTH: 7412             CTF: 0              CSZ: 8388608
UTE: 241381           UTH: 505258           UTF: 0              SBA: 15
BAS: 44               RED: 45               LEN: 0+19=19   OPT: k1 m1 z0 h1 n1 b1


----------------------------------------------------------------------
                         MODIFY THEOREM

Nodes searched last time = 32985    Time taken = 5    Time to go = 395
Maxtime on this pass = 197

Single weakening proof found => This time none.
----------------------------------------------------------------------

   0 <BC: 44 NC: 3 AC: 52 U: 67>
1 {T0 N2506-2506 [C7495 H86 h0 U75802 u142962 f0] E3 s807 t0 L0}
2 {T3 N19951-19951 [C62431 H7783 h0 U429436 u1610371 f0] E10 s9326 t0 L0}*

Proof Found; details coming!

1 {T4 N22457-2506 [C7495 H86 h0 U75802 u142962 f0] E3 s10133 t0 L0}
2 {T7 N37170-17219 [C51166 H6490 h0 U381430 u1423276 f0] E10 s15864 t0 L0}
1 {T8 N39676-2506 [C7495 H86 h0 U75802 u142962 f0] E3 s16671 t0 L0}
2 {T11 N54365-17195 [C51101 H6488 h0 U381327 u1423143 f0] E10 s22394 t0 L0}
1 {T12 N56871-2506 [C7495 H86 h0 U75802 u142962 f0] E3 s23201 t0 L0}
2 {T15 N71191-16826 [C49847 H6367 h0 U379313 u1337297 f0] E10 s28325 t0 L0}
1 {T15 N73697-2506 [C7495 H86 h0 U75802 u142962 f0] E3 s29132 t0 L0}
2 {T16 N76991-5800 [C17890 H2765 h0 U79677 u184987 f0] E3 s29959 t0 L0}$

THE PROOF

Axioms:
 1: E.[x,f3[y,z]] ~p1[SK2[y,z,x],x] ~p1[SK2[y,z,x],y] ~p1[SK2[y,z,x],z]
 2: E.[x,f2[y,z]] E.[SK1[y,z,x],y] E.[SK1[y,z,x],z] p1[SK1[y,z,x],x]
 3S>~E.[x,f3[y,z]] p1[u,x] ~p1[u,y] ~p1[u,z]
 4 >~E.[x,f2[y,z]] ~p1[u,x] E.[u,y] E.[u,z]
 5: E.[x,f3[y,z]] p1[SK2[y,z,x],z] p1[SK2[y,z,x],x]
 6: E.[x,f3[y,z]] p1[SK2[y,z,x],y] p1[SK2[y,z,x],x]
 7: E.[x,f2[y,z]] ~p1[SK1[y,z,x],x] ~E.[SK1[y,z,x],z]
 8: E.[x,f2[y,z]] ~p1[SK1[y,z,x],x] ~E.[SK1[y,z,x],y]
 9S>E.[x,f4[y]] E.[SK3[y,x],y] p1[SK3[y,x],x]
10S>E.[x,f4[y]] ~p1[SK3[y,x],x] ~E.[SK3[y,x],y]
11S>~E.[x,f3[y,z]] ~p1[u,x] p1[u,z]
12: ~E.[x,f3[y,z]] ~p1[u,x] p1[u,y]
13 >~E.[x,f2[y,z]] p1[u,x] ~E.[u,z]
14: ~E.[x,f2[y,z]] p1[u,x] ~E.[u,y]
15S>~E.[x,f4[y]] p1[z,x] ~E.[z,y]
16S ~E.[x,f4[y]] ~p1[z,x] E.[z,y]
17 >~E.[x,y] ~E.[y,z] E.[x,z]
18: ~p1[x,y] ~E.[y,z] p1[x,z]
19 >~p1[x,y] ~E.[x,z] p1[z,y]
20: ~p5[x] ~E.[x,y] p5[y]
21: ~E.[x,y] E.[SK1[x,z,u],SK1[y,z,u]]
22: ~E.[x,y] E.[SK1[z,x,u],SK1[z,y,u]]
23: ~E.[x,y] E.[SK1[z,u,x],SK1[z,u,y]]
24: ~E.[x,y] E.[SK2[x,z,u],SK2[y,z,u]]
25: ~E.[x,y] E.[SK2[z,x,u],SK2[z,y,u]]
26: ~E.[x,y] E.[SK2[z,u,x],SK2[z,u,y]]
27: ~E.[x,y] E.[f3[z,x],f3[z,y]]
28 >~E.[x,y] E.[f3[x,z],f3[y,z]]
29 >~E.[x,y] E.[f2[z,x],f2[z,y]]
30: ~E.[x,y] E.[f2[x,z],f2[y,z]]
31: ~E.[x,y] E.[SK3[x,z],SK3[y,z]]
32 >~E.[x,y] E.[SK3[z,x],SK3[z,y]]
33: ~E.[x,y] E.[f4[x],f4[y]]
34: ~p1[x,y] ~p1[y,x]
35 >~E.[x,y] E.[y,x]
36S>E.[f3[x,y],f3[y,x]]
37 >E.[f2[x,y],f2[y,x]]
38S>E.[f3[x,x],x]
39S>E.[x,x]
40: p5[SK5]
```

```
41: ~p5[SK4]

Negated conclusion:
42S>~p1[SK8,SK7] E.[SK6,SK8]
43S>~E.[f3[f2[SK6,SK8],SK7],f4[SK6]]
44S>p1[SK6,SK7]

---------------

Weakened clauses used in proof:
51# (43a,9a)  E.[SK3[SK6,f3[f2[SK6,SK8],SK7]],SK6]
p1[SK3[SK6,f3[f2[SK6,SK8],SK7]],f3[f2[SK6,SK8],SK7]]
52# (51b,11b) E.[SK3[SK6,f3[f2[SK6,SK8],SK7]],SK6] ~E.[f3[f2[SK6,SK8],SK7],f3[x,y]]
p1[SK3[SK6,f3[f2[SK6,SK8],SK7]],y]
53: (52b,36a) E.[SK3[SK6,f3[f2[SK6,SK8],SK7]],SK6]
p1[SK3[SK6,f3[f2[SK6,SK8],SK7]],f2[SK6,SK8]]
62# (43a,10a) ~p1[SK3[SK6,f3[f2[SK6,SK8],SK7]],f3[f2[SK6,SK8],SK7]]
~E.[SK3[SK6,f3[f2[SK6,SK8],SK7]],SK6]
63# (62b,16c) ~p1[SK3[SK6,f3[f2[SK6,SK8],SK7]],f3[f2[SK6,SK8],SK7]] ~E.[x,f4[SK6]]
~p1[SK3[SK6,f3[f2[SK6,SK8],SK7]],x]
64# (63b,38a) ~p1[SK3[SK6,f3[f2[SK6,SK8],SK7]],f3[f2[SK6,SK8],SK7]]
~p1[SK3[SK6,f3[f2[SK6,SK8],SK7]],f3[f4[SK6],f4[SK6]]]
65: (64b,11c) ~p1[SK3[SK6,f3[f2[SK6,SK8],SK7]],f3[f2[SK6,SK8],SK7]]
~E.[x,f3[y,f3[f4[SK6],f4[SK6]]]] ~p1[SK3[SK6,f3[f2[SK6,SK8],SK7]],x]

Phase 0 clauses used in proof:
47S>E.[SK3[SK6,f3[f2[SK6,SK8],SK7]],SK6] p1[SK3[SK6,f3[f2[SK6,SK8],SK7]],f2[SK6,SK8]]
(Clause 53 in w-proof)
49 >~p1[SK3[SK6,f3[f2[SK6,SK8],SK7]],f3[f2[SK6,SK8],SK7]]
~E.[x,f3[y,f3[f4[SK6],f4[SK6]]]] ~p1[SK3[SK6,f3[f2[SK6,SK8],SK7]],x]
(Clause 65 in w-proof)

Phases 1 and 2 clauses used in proof:
58S>(44a,3d)  ~E.[x,f3[y,SK7]] p1[SK6,x] ~p1[SK6,y]
 59S>(58a,39a) p1[SK6,f3[x,SK7]] ~p1[SK6,x]
  60S>(59b,13b) p1[SK6,f3[x,SK7]] ~E.[x,f2[y,z]] ~E.[SK6,z]
   61S>(60b,37a) p1[SK6,f3[f2[x,y],SK7]] ~E.[SK6,x]
    62S>(61b,39a) p1[SK6,f3[f2[SK6,y],SK7]]

63S>(43a,9a)  E.[SK3[SK6,f3[f2[SK6,SK8],SK7]],SK6]
p1[SK3[SK6,f3[f2[SK6,SK8],SK7]],f3[f2[SK6,SK8],SK7]]
 64S>(63a,35a) p1[SK3[SK6,f3[f2[SK6,SK8],SK7]],f3[f2[SK6,SK8],SK7]]
E.[SK6,SK3[SK6,f3[f2[SK6,SK8],SK7]]]
  65S>(64b,19b) p1[SK3[SK6,f3[f2[SK6,SK8],SK7]],f3[f2[SK6,SK8],SK7]] ~p1[SK6,x]
p1[SK3[SK6,f3[f2[SK6,SK8],SK7]],x]
   66S>(65ac)    ~p1[SK6,f3[f2[SK6,SK8],SK7]]
p1[SK3[SK6,f3[f2[SK6,SK8],SK7]],f3[f2[SK6,SK8],SK7]]
    67S>(66a,62a) p1[SK3[SK6,f3[f2[SK6,SK8],SK7]],f3[f2[SK6,SK8],SK7]]

68S>(43a,10a) ~p1[SK3[SK6,f3[f2[SK6,SK8],SK7]],f3[f2[SK6,SK8],SK7]]
~E.[SK3[SK6,f3[f2[SK6,SK8],SK7]],SK6]
 69S>(68b,17c) ~p1[SK3[SK6,f3[f2[SK6,SK8],SK7]],f3[f2[SK6,SK8],SK7]]
~E.[SK3[SK6,f3[f2[SK6,SK8],SK7]],x] ~E.[x,SK6]
  70S>(69a,67a) ~E.[SK3[SK6,f3[f2[SK6,SK8],SK7]],x] ~E.[x,SK6]
   71S>(70a,32b) ~E.[SK3[SK6,x],SK6] ~E.[f3[f2[SK6,SK8],SK7],x]
    72S>(71b,36a) ~E.[SK3[SK6,f3[SK7,f2[SK6,SK8]]],SK6]

73S>(43a,10a) ~p1[SK3[SK6,f3[f2[SK6,SK8],SK7]],f3[f2[SK6,SK8],SK7]]
~E.[SK3[SK6,f3[f2[SK6,SK8],SK7]],SK6]
 74S>(73b,17c) ~p1[SK3[SK6,f3[f2[SK6,SK8],SK7]],f3[f2[SK6,SK8],SK7]]
~E.[SK3[SK6,f3[f2[SK6,SK8],SK7]],x] ~E.[x,SK6]
  75S>(74a,67a) ~E.[SK3[SK6,f3[f2[SK6,SK8],SK7]],x] ~E.[x,SK6]
   76S>(75b,39a) ~E.[SK3[SK6,f3[f2[SK6,SK8],SK7]],SK6]

77S>(39a,4a)  ~p1[x,f2[y,z]] E.[x,y] E.[x,z]
 78S>(77a*47b) E.[SK3[SK6,f3[f2[SK6,SK8],SK7]],SK6] E.[SK3[SK6,f3[f2[SK6,SK8],SK7]],SK8]
  79S>(78a,76a) E.[SK3[SK6,f3[f2[SK6,SK8],SK7]],SK8]

80S>(36a,32a) E.[SK3[x,f3[y,z]],SK3[x,f3[z,y]]]
 81S>(80a,17a) ~E.[SK3[x,f3[y,z]],u] E.[SK3[x,f3[z,y]],u]
  82S>(81a,9b)  E.[SK3[x,f3[y,z]],x] E.[f3[z,y],f4[x]] p1[SK3[x,f3[z,y]],f3[z,y]]
```

```
   83S>(82b,43a) E.[SK3[SK6,f3[SK7,f2[SK6,SK8]]],SK6]
p1[SK3[SK6,f3[f2[SK6,SK8],SK7]],f3[f2[SK6,SK8],SK7]]
    84S>[83b,79a] E.[SK3[SK6,f3[SK7,f2[SK6,SK8]]],SK6] p1[SK8,f3[f2[SK6,SK8],SK7]]
     85S>(84a,72a) p1[SK8,f3[f2[SK6,SK8],SK7]]

86S>(36a,17b) ~E.[x,f3[y,z]] E.[x,f3[z,y]]
 87S>(86b,35a) ~E.[x,f3[y,z]] E.[f3[z,y],x]
  88S>(87b,49b) ~E.[f3[x,f3[f4[SK6],f4[SK6]]],f3[y,z]]
~p1[SK3[SK6,f3[f2[SK6,SK8],SK7]],f3[f2[SK6,SK8],SK7]]
~p1[SK3[SK6,f3[f2[SK6,SK8],SK7]],f3[z,y]]
   89S>[88b,79a] ~E.[f3[x,f3[f4[SK6],f4[SK6]]],f3[y,z]] ~p1[SK8,f3[f2[SK6,SK8],SK7]]
~p1[SK3[SK6,f3[f2[SK6,SK8],SK7]],f3[z,y]]
    90S>[89c,79a] ~E.[f3[x,f3[f4[SK6],f4[SK6]]],f3[y,z]] ~p1[SK8,f3[f2[SK6,SK8],SK7]]
~p1[SK8,f3[z,y]]
     91S>(90b,85a) ~E.[f3[x,f3[f4[SK6],f4[SK6]]],f3[y,z]] ~p1[SK8,f3[z,y]]
      92S>(91a,38a) ~p1[SK8,f3[f4[SK6],f4[SK6]]]

93S>(36a,17a) ~E.[f3[x,y],z] E.[f3[y,x],z]
 94S>(93b,15a) ~E.[f3[x,y],f4[z]] p1[u,f3[y,x]] ~E.[u,z]
  95S>(94a,38a) p1[x,f3[f4[y],f4[y]]] ~E.[x,y]
   96S>(95a,92a) ~E.[SK8,SK6]

97S>(36a,17a) ~E.[f3[x,y],z] E.[f3[y,x],z]
 98S>(97b,17a) ~E.[f3[x,y],z] ~E.[z,u] E.[f3[y,x],u]
  99S>(98b,35b) ~E.[f3[x,y],z] E.[f3[y,x],u] ~E.[u,z]
   100S>(99a,38a) E.[f3[x,x],y] ~E.[y,x]
    101S>[100a,38a] E.[x,y] ~E.[y,x]
     102S>(101a,96a) ~E.[SK6,SK8]

103S>(28b,11a) ~E.[x,y] ~p1[z,f3[x,u]] p1[z,u]
 104S>(103a,29b) ~p1[x,f3[f2[y,z],u]] p1[x,u] ~E.[z,v]
  105S>(104b,42a) ~p1[SK8,f3[f2[x,y],SK7]] ~E.[y,z] E.[SK6,SK8]
   106S>(105c,102a) ~p1[SK8,f3[f2[x,y],SK7]] ~E.[y,z]
    107S>(106b,39a) ~p1[SK8,f3[f2[x,y],SK7]]
     108S>(107a,85a) []


END OF PROOF

PHASE 0: 0 s, PHASE 1: 3 s, PHASE 2: 13 s ACCUM: 5  Total Time: 21 s
NOD: 76991            RES: 261490           FAC: 13889            T: 16  V: 8  L: 3
CTE: 62431            CTH: 7783             CTF: 0                CSZ: 8388608
UTE: 429436           UTH: 1610371          UTF: 0                SBA: 10
BAS: 44               RED: 52               LEN: 2+51=53   OPT: k1 m1 z0 h1 n1 b1
t400/395
```

APPENDIX 3: OCTOPUS FOF RESULTS (RUN 1) CASC 2006

THM	NAME	FROM	WEAK	PICK	BASE	SPLT	LITS	VARS	TRMS	KNOW	TIME	LENG
116	AGT005+2	M.	1606.	NA.	1003.	0.	5.	10.	10.	6.	6.	4.
138	AGT007+2	9.	1606.	4l	1003.	0.	5.	10.	10.	161.	407.	41.
102	AGT015+1	1.	1528.	0l	673.	0.	5.	10.	10.	-1.	7.	3.
131	AGT018+1	104.	1606.	88l	673.	0.	4.	10.	9.	383.	421.	28.
84	AGT024+2	M.	1606.	NA.	1003.	0.	5.	10.	10.	52.	93.	26.
25	ALG074+1	79.	87.	63f	24.	0.	5.	6.	20.	3.	4.	73.
28	ALG211+1	M.	53.	NA.	10.	0.	3.	6.	12.	6.	6.	10.
103	CSR001+1	FAIL.	1030.	NA.	201.	0.						
86	CSR002+1	FAIL.	1029.	NA.	201.	0.						
109	CSR005+2	FAIL.	1031.	NA.	202.	0.						
93	CSR007+1	52.	1030.	-167f	201.	0.	8.	10.	11.	153.	175.	80.
111	CSR019+1	1.	1417.	0c	256.	0.	6.	10.	10.	5.	6.	26.
119	CSR021+1	1.	1417.	0c	256.	0.	5.	10.	10.	-1.	7.	13.
39	CSR022+1	1.	1417.	0c	256.	0.	5.	10.	10.	-1.	11.	13.
81	CSR024+1	FAIL.										
82	GEO085+1	1.	312.	0c	87.	0.	5.	10.	10.	-1.	6.	6.
123	GEO088+1	3.	312.	1c	89.	0.	7.	10.	13.	4.	5.	35.
129	GEO109+1	FAIL.	358.	NA.	104.	-: 9.						
51	GRA002+1	FAIL.	509.	NA.	121.	0.						
142	GRA002+2	FAIL.	539.	NA.	129.	0.						
104	GRA002+4	4.	516.	1c	122.	0.	7.	12.	7.	-1.	5.	9.
72	GRA004+1	16.	505.	7c	122.	7: 9.	3.	12.	6.	-1.	8.	5.
	GRA004+1	14.	505.	6l	122.	5: 9.	10.	12.	6.	-1.	8.	28.
	GRA004+1	6.	505.	2l	122.	6: 9.	10.	12.	6.	-1.	8.	28.
	GRA004+1	20.	505.	9l	122.	2: 9.	7.	12.	6.	-1.	8.	28.
	GRA004+1	19.	505.	9l	122.	1: 9.	6.	12.	10.	-1.	9.	5.
	GRA004+1	22.	505.	10l	122.	4: 9.	10.	12.	10.	-1.	9.	5.
	GRA004+1	36.	505.	20l	122.	9: 9.	7.	12.	7.	-1.	9.	4.
	GRA004+1	21.	505.	10l	122.	3: 9.	8.	12.	10.	-1.	9.	5.
	GRA004+1	35.	505.	19l	122.	8: 9.	6.	12.	11.	-1.	9.	28.
45	GRA006+1	FAIL.	621.	NA.	133.	-:10.						
110	GRA007+1	66.	516.	50l	122.	0.	6.	12.	8.	218.	443.	105.
88	GRA007+2	66.	526.	50l	123.	0.	6.	12.	8.	264.	535.	105.
19	GRA008+1	FAIL.	509.	NA.	123.	0.						
120	GRA009+2	FAIL.	510.	NA.	124.	0.						
139	GRA010+1	5.	519.	2c	124.	0.	8.	12.	11.	-1.	1.	12.
96	GRA011+1	FAIL.	506.	NA.	121.	0.						
108	GRA012+1	FAIL.	507.	NA.	121.	0.						
38	GRP001+6	M.	41.	NA.	9.	0.	3.	12.	9.	2.	3.	13.
2	GRP012+5	M.	46.	NA.	9.	0.	3.	12.	9.	3.	3.	13.
30	LCL414+1	1.	29.	0c	4.	0.	5.	10.	27.	-1.	8.	5.
52	MGT022+1	M.	43.	NA.	9.	-:10.	4.	8.	14.	5.	5.	22.
54	MGT022+2	M.	43.	NA.	9.	-:10.	4.	8.	14.	4.	4.	22.
56	MGT028+1	M.	156.	NA.	13.	0.	6.	6.	10.	6.	6.	44.
44	MGT036+1	M.	91.	NA.	13.	-: 2.	4.	8.	9.	8.	8.	18.
34	MGT036+3	M.	59.	NA.	8.	0.	4.	8.	9.	6.	6.	9.
132	MGT042+1	28.	249.	13l	57.	c:10.	7.	6.	10.	-1.	2.	12.
	MGT042+1	30.	249.	14c	57.	e:10.	10.	6.	7.	-1.	2.	12.
	MGT042+1	34.	249.	18c	57.	2:10.	5.	6.	7.	-1.	2.	12.
	MGT042+1	36.	249.	20c	57.	4:10.	7.	6.	7.	-1.	2.	12.
	MGT042+1	38.	249.	22c	57.	6:10.	10.	6.	11.	-1.	2.	12.
	MGT042+1	42.	249.	26c	57.	a:10.	5.	6.	11.	-1.	2.	12.
	MGT042+1	49.	249.	33l	57.	1:10.	5.	6.	11.	-1.	2.	29.
	MGT042+1	50.	249.	34c	57.	3:10.	5.	6.	7.	-1.	3.	41.
	MGT042+1	9.	249.	4l	57.	9:10.	4.	6.	10.	-1.	3.	42.
	MGT042+1	43.	249.	27l	57.	b:10.	7.	6.	14.	-1.	4.	51.
	MGT042+1	44.	249.	28c	57.	d:10.	7.	6.	7.	-1.	6.	40.
	MGT042+1	54.	249.	38c	57.	7:10.	10.	6.	7.	-1.	6.	40.
	MGT042+1	53.	249.	37l	57.	5:10.	8.	6.	14.	-1.	7.	41.
	MGT042+1	46.	249.	30c	57.	f:10.	10.	6.	7.	-1.	7.	40.
	MGT042+1	38.	249.	22c	57.	8:10.	10.	6.	11.	-1.	12.	120.

THM	NAME	FROM	WEAK	PICK	BASE	SPLT	LITS	VARS	TRMS	KNOW	TIME	LENG
	MGT042+1	62.	249.	46c	57.	10:10.	10.	6.	12.	-1.	12.	25.
99	MGT060+1	9.	107.	4c	44.	0.	4.	6.	14.	6.	6.	54.
114	MSC010+1	FAIL.										
137	NUM310+1	M.	1533.	NA.	438.	0.	7.	22.	10.	12.	30.	20.
117	NUM327+1	M.	1533.	NA.	438.	0.	7.	22.	10.	7.	7.	9.
147	NUM329+1	M.	1533.	NA.	438.	0.	7.	22.	10.	7.	8.	9.
73	NUM334+1	M.	1533.	NA.	438.	0.	7.	22.	10.	7.	9.	8.
130	NUM398+1	15.	166.	7l	101.	0.	8.	6.	10.	2.	2.	21.
76	NUM400+1	M.	221.	NA.	127.	-: 8.	3.	8.	12.	30.	47.	52.
20	PUZ047+1	M.	173.	NA.	15.	0.	3.	6.	14.	7.	7.	8.
95	SET008+3	1.	86.	0c	38.	0.	4.	6.	14.	3.	3.	12.
50	SET009+3	58.	36.	6c	8.	0.	5.	6.	15.	-1.	4.	14.
59	SET014+3	3.	58.	1c	27.	0.	6.	6.	16.	-1.	7.	9.
11	SET067+1	M.	488.	NA.	133.	0.	3.	8.	18.	5.	6.	11.
98	SET094+1	M.	489.	NA.	136.	0.	3.	8.	16.	1.	2.	9.
80	SET099+1	FAIL.	493.	NA.	135.	0.						
122	SET144+3	28.	89.	13c	35.	0.	4.	9.	24.	6.	7.	39.
146	SET171+3	57.	89.	41f	34.	0.	4.	6.	33.	6.	21.	80.
143	SET200+3	M.	60.	NA.	27.	0.	3.	6.	14.	7.	7.	11.
150	SET577+3	65.	81.	49f	31.	0.	4.	6.	14.	3.	3.	17.
140	SET579+3	29.	59.	14l	24.	0.	4.	6.	12.	1.	1.	24.
128	SET582+3	56.	139.	40l	45.	0.	3.	6.	12.	7.	21.	62.
41	SET584+3	1.	58.	0c	26.	0.	5.	6.	14.	-1.	5.	9.
141	SET587+3	2.	82.	49f	37.	2: 4.	4.	6.	12.	-1.	6.	1.
	SET587+3	7.	82.	3l	37.	3: 4.	3.	6.	16.	-1.	6.	1.
	SET587+3	4.	82.	50v	37.	4: 4.	7.	6.	8.	-1.	7.	9.
	SET587+3	45.	82.	29f	37.	1: 4.	5.	6.	13.	-1.	7.	19.
6	SET588+3	M.	36.	NA.	8.	0.	3.	6.	14.	7.	7.	14.
48	SET589+3	M.	54.	NA.	12.	0.	3.	6.	14.	6.	6.	9.
36	SET590+3	1.	32.	0c	7.	0.	5.	6.	10.	-1.	4.	7.
53	SET593+3	7.	68.	3c	31.	1: 2.	3.	6.	16.	-1.	13.	12.
	SET593+3	4.	68.	1c	31.	2: 2.	6.	6.	8.	-1.	15.	12.
97	SET594+3	5.	91.	2c	35.	0.	7.	6.	18.	-1.	6.	12.
133	SET595+3	49.	99.	33f	38.	0.	4.	6.	15.	3.	3.	17.
127	SET603+3	1.	84.	0c	36.	0.	5.	6.	12.	-1.	6.	8.
136	SET623+3	FAIL.	122.	NA.	40.	0.						
75	SET624+3	40.	53.	24f	27.	4: c.	7.	6.	9.	-1.	5.	5.
	SET624+3	41.	53.	25f	27.	5: c.	8.	6.	13.	-1.	5.	7.
	SET624+3	44.	53.	28f	27.	8: c.	4.	6.	9.	-1.	5.	5.
	SET624+3	42.	53.	26f	27.	6: c.	3.	6.	13.	-1.	6.	7.
	SET624+3	2.	53.	0l	27.	2: c.	4.	6.	12.	-1.	6.	1.
	SET624+3	47.	53.	31f	27.	b: c.	7.	6.	17.	-1.	6.	5.
	SET624+3	57.	53.	41f	27.	9: c.	3.	6.	17.	-1.	6.	5.
	SET624+3	58.	53.	42f	27.	a: c.	5.	6.	13.	-1.	6.	5.
	SET624+3	60.	53.	44v	27.	c: c.	6.	6.	10.	-1.	6.	10.
	SET624+3	19.	53.	9f	27.	7: c.	7.	6.	12.	-1.	6.	5.
	SET624+3	61.	53.	45f	27.	1: c.	7.	6.	14.	-1.	6.	7.
	SET624+3	63.	53.	47f	27.	3: c.	7.	6.	18.	-1.	6.	7.
10	SET631+3	29.	38.	14v	8.	0.	3.	6.	10.	-1.	6.	7.
135	SET655+3	17.	180.	-139v	77.	0.	5.	8.	13.	2.	3.	17.
105	SET656+3	1.	392.	0c	119.	0.	5.	10.	11.	-1.	3.	10.
61	SET657+3	M.	321.	NA.	114.	0.	6.	10.	10.	1.	2.	31.
35	SET669+3	1.	322.	0c	110.	1: 2.	5.	8.	10.	-1.	4.	7.
	SET669+3	58.	322.	42l	110.	2: 2.	6.	8.	11.	-1.	6.	18.
134	SET680+3	M.	310.	NA.	110.	-:10.	6.	10.	10.	8.	8.	28.
21	SET795+4	26.	1540.	12c	240.	0.	6.	10.	7.	19.	22.	22.
15	SET796+4	FAIL.	1719.	NA.	367.	0.						
121	SET798+4	4.	1539.	1c	241.	0.	7.	10.	6.	-1.	5.	7.
145	SET802+4	4.	1529.	1c	240.	4: c.	7.	10.	6.	-1.	11.	1.
	SET802+4	14.	1529.	-137v	240.	2: c.	10.	10.	6.	-1.	11.	1.
	SET802+4	6.	1529.	2c	240.	6: c.	10.	10.	6.	-1.	11.	2.
	SET802+4	8.	1529.	3l	240.	8: c.	3.	10.	10.	-1.	11.	2.
	SET802+4	5.	1529.	2c	240.	5: c.	8.	10.	10.	-1.	11.	8.
	SET802+4	7.	1529.	3l	240.	7: c.	8.	10.	13.	-1.	11.	5.
	SET802+4	27.	1529.	-144v	240.	3: c.	6.	10.	13.	-1.	11.	1.
	SET802+4	58.	1529.	42l	240.	a: c.	5.	10.	11.	-1.	12.	1.
	SET802+4	57.	1529.	41l	240.	9: c.	4.	10.	14.	-1.	12.	1.

THM	NAME	FROM	WEAK	PICK	BASE	SPLT	LITS	VARS	TRMS	KNOW	TIME	LENG
	SET802+4	1.	1529.	0c	240.	1: c.	5.	10.	10.	-1.	13.	12.
	SET802+4	60.	1529.	44l	240.	c: c.	8.	10.	8.	-1.	13.	13.
	SET802+4	95.	1529.	79l	240.	b: c.	9.	10.	13.	-1.	27.	13.
65	SET805+4	FAIL.	1531.	NA.	238.	0.						
5	SET806+4	FAIL.	1522.	NA.	202.	0.						
23	SET808+4	FAIL.	589.	NA.	143.	0.						
31	SET811+4	104.	590.	88f	143.	0.	4.	10.	9.	17.	32.	26.
74	SET812+4	54.	588.	38l	142.	0.	11.	10.	7.	19.	29.	20.
69	SET814+4	1.	590.	0c	143.	0.	5.	10.	10.	-1.	2.	9.
9	SET815+4	FAIL.	591.	NA.	143.	0.						
112	SET816+4	FAIL.	590.	NA.	143.	0.						
68	SET896+1	M.	13.	NA.	7.	-: 4.	3.	4.	10.	6.	6.	4.
64	SET899+1	M.	21.	NA.	7.	0.	3.	6.	10.	6.	6.	3.
26	SET913+1	M.	17.	NA.	6.	0.	3.	4.	10.	5.	5.	2.
46	SET915+1	M.	17.	NA.	6.	0.	3.	4.	10.	7.	7.	2.
92	SET919+1	M.	169.	NA.	43.	-: 2.	3.	8.	16.	3.	6.	40.
13	SET935+1	30.	141.	14l	46.	0.	5.	8.	13.	2.	2.	30.
22	SET939+1	M.	14.	NA.	6.	0.	3.	4.	10.	6.	6.	2.
125	SET942+1	1.	104.	0c	34.	0.	5.	8.	14.	-1.	4.	12.
87	SET951+1	132.	271.	116f	63.	0.	9.	12.	24.	30.	41.	50.
42	SET966+1	4.	28.	1v	8.	0.	6.	6.	9.	-1.	4.	5.
91	SET982+1	FAIL.	346.	NA.	55.	0.						
100	SEU003+1	107.	300.	91c	96.	0.	7.	8.	16.	84.	84.	24.
148	SEU009+1	M.	192.	NA.	80.	-: c.	6.	6.	10.	8.	11.	25.
118	SEU014+1	FAIL.	204.	NA.	84.	0.						
67	SEU075+1	BUG.										
17	SEU086+1	FAIL.	399.	NA.	213.	0.						
107	SWC078+1	FAIL.	1230.	NA.	289.	-:10.						
115	SWC231+1	FAIL.	1317.	NA.	295.	0.						
29	SWC239+1	24.	1310.	11l	298.	8:10.	3.	12.	8.	-1.	3.	9.
	SWC239+1	9.	1310.	4l	298.	9:10.	4.	12.	12.	-1.	3.	11.
	SWC239+1	49.	1310.	33l	298.	1:10.	4.	12.	13.	-1.	4.	11.
	SWC239+1	10.	1310.	4l	298.	a:10.	5.	12.	8.	-1.	4.	20.
	SWC239+1	66.	1310.	50l	298.	2:10.	5.	12.	10.	-1.	5.	20.
	SWC239+1	72.	1310.	56l	298.	b:10.	3.	12.	14.	-1.	9.	9.
	SWC239+1	66.	1310.	50l	298.	3:10.	5.	12.	10.	-1.	10.	20.
	SWC239+1	72.	1310.	56l	298.	c:10.	3.	12.	14.	-1.	13.	9.
	SWC239+1	72.	1310.	56l	298.	d:10.	3.	12.	14.	-1.	17.	9.
	SWC239+1	66.	1310.	50l	298.	4:10.	5.	12.	10.	-1.	20.	20.
	SWC239+1	6.	1310.	2l	298.	6:10.	10.	12.	8.	-1.	20.	20.
	SWC239+1	72.	1310.	56l	298.	e:10.	3.	12.	14.	-1.	21.	9.
	SWC239+1	72.	1310.	56l	298.	f:10.	3.	12.	14.	-1.	25.	9.
	SWC239+1	49.	1310.	33l	298.	5:10.	4.	12.	13.	-1.	27.	11.
	SWC239+1	82.	1310.	66l	298.	7:10.	6.	12.	14.	-1.	35.	56.
	SWC239+1	72.	1310.	56l	298.	10:10.	4.	12.	14.	-1.	44.	25.
66	SWV011+1	M.	97.	NA.	16.	0.	3.	2.	35.	5.	5.	2.
18	SWV014+1	6.	216.	2l	37.	0.	9.	14.	14.	1.	1.	28.
113	SWV026+1	FAIL.	2000.	NA.	467.	-: 7.						
55	SWV030+1	6.	913.	2v	198.	6:10.	11.	24.	6.	-1.	3.	14.
	SWV030+1	26.	913.	12l	198.	a:10.	6.	24.	6.	-1.	4.	10.
	SWV030+1	4.	913.	1l	198.	4:10.	8.	24.	6.	-1.	4.	14.
	SWV030+1	41.	913.	25l	198.	9:10.	4.	24.	11.	-1.	5.	3.
	SWV030+1	30.	913.	14l	198.	e:10.	10.	24.	7.	-1.	5.	16.
	SWV030+1	49.	913.	33l	198.	1:10.	4.	24.	11.	-1.	6.	3.
	SWV030+1	2.	913.	-131l	198.	2:10.	5.	24.	10.	-1.	6.	3.
	SWV030+1	4.	913.	1l	198.	5:10.	8.	24.	6.	-1.	7.	13.
	SWV030+1	26.	913.	12l	198.	b:10.	6.	24.	6.	-1.	8.	17.
	SWV030+1	6.	913.	2v	198.	7:10.	11.	24.	6.	-1.	10.	21.
	SWV030+1	60.	913.	44c	198.	c:10.	7.	24.	8.	-1.	11.	16.
	SWV030+1	30.	913.	14l	198.	f:10.	11.	24.	7.	-1.	11.	24.
	SWV030+1	34.	913.	18c	198.	3:10.	6.	24.	7.	-1.	13.	17.
	SWV030+1	6.	913.	2v	198.	8:10.	11.	24.	6.	-1.	14.	16.
	SWV030+1	26.	913.	12l	198.	d:10.	6.	24.	6.	-1.	15.	17.
	SWV030+1	26.	913.	12l	198.	10:10.	6.	24.	6.	-1.	60.	16.
106	SWV033+1	FAIL.	880.	NA.	177.	-:10.						
43	SWV036+1	FAIL.										
77	SWV038+1	FAIL.										

THM	NAME	FROM	WEAK	PICK	BASE	SPLT	LITS	VARS	TRMS	KNOW	TIME	LENG
47	SWV039+1	FAIL.	2000.	NA.	442.	-: 7.						
149	SWV041+1	56.	809.	40c	179.	0.	3.	24.	7.	1.	1.	8.
63	SWV048+1	114.	805.	98f	172.	0.	6.	24.	9.	6.	6.	19.
33	SWV050+1	13.	821.	-137f	163.	0.	9.	24.	13.	236.	246.	19.
101	SWV053+1	FAIL.	1134.	NA.	184.	-: 9.						
90	SWV090+1	FAIL.	814.	NA.	173.	-: 5.						
71	SWV092+1	6.	872.	2c	189.	2: 2.	10.	24.	6.	-1.	1.	8.
	SWV092+1	4.	872.	1c	189.	1: 2.	7.	24.	6.	-1.	2.	8.
37	SWV094+1	FAIL.	1251.	NA.	244.	-: c.						
94	SWV096+1	10.	976.	4c	212.	2: 2.	5.	24.	6.	-1.	2.	5.
	SWV096+1	10.	976.	4c	212.	1: 2.	5.	24.	6.	-1.	2.	5.
57	SWV098+1	24.	962.	11l	202.	2: 2.	4.	24.	6.	-1.	2.	12.
	SWV098+1	24.	962.	11l	202.	1: 2.	4.	24.	6.	-1.	8.	12.
85	SWV103+1	FAIL.	778.	NA.	164.	-: 5.						
89	SWV117+1	FAIL.	2000.	NA.	805.	-:10.						
144	SWV161+1	64.	802.	48l	171.	0.	3.	24.	8.	3.	3.	11.
78	SWV173+1	10.	864.	4l	178.	0.	5.	24.	6.	-1.	2.	3.
7	SWV177+1	10.	870.	4l	180.	0.	5.	24.	6.	-1.	2.	3.
126	SWV200+1	M.	928.	NA.	204.	-: 8.	11.	24.	10.	6.	6.	5.
83	SWV233+1	FAIL.	171.	NA.	60.	0.						
124	SWV234+1	FAIL.	165.	NA.	35.	0.						
1	SWV234+2	FAIL.	106.	NA.	31.	0.						
49	SWV235+1	4.	182.	1c	38.	0.	6.	12.	12.	-1.	1.	25.
27	SWV236+1	24.	177.	-142f	37.	0.	6.	12.	12.	1.	1.	10.
79	SWV237+1	M.	130.	NA.	31.	-: c.	3.	6.	16.	5.	5.	9.
58	SWV938+1	FAIL.										
70	SYN081+1	M.	11.	NA.	2.	0.	3.	2.	10.	6.	6.	3.
4	SYN084+1	M.	266.	NA.	30.	-:10.	4.	4.	9.	1.	1.	32.
14	SYN364+1	M.	17.	NA.	4.	-: 4.	3.	6.	10.	5.	5.	9.
12	SYN728+1	M.	17.	NA.	4.	-: 4.	3.	6.	10.	5.	5.	9.
62	SYN730+1	M.	7.	NA.	1.	0.	3.	4.	14.	6.	6.	1.
60	SYN939+1	M.	10.	NA.	4.	-: c.	3.	4.	9.	8.	8.	11.
32	SYN940+1	M.	1.	NA.	3.	-: c.	3.	4.	9.	9.	9.	7.
8	SYN941+1	M.	2.	NA.	3.	-: c.	3.	4.	9.	7.	7.	7.
24	SYN943+1	M.	25.	NA.	6.	0.	3.	2.	10.	4.	4.	8.
40	SYN980+1	M.	9.	NA.	3.	-: 4.	3.	4.	10.	4.	4.	7.

APPENDIX 4: OCTOPUS FOF RESULTS (RUN 2) CASC 2006

THM	NAME	FROM	WEAK	PICK	BASE	SPLT	LITS	VARS	TRMS	KNOW	TIME	LENG
116	AGT005+2	M.	1606.	NA.	1003.	0.	5.	10.	10.	4.	4.	4.
138	AGT007+2	44.	1606.	-159f	1003.	0.	8.	10.	7.	154.	271.	43.
102	AGT015+1	1.	1528.	0l	673.	0.	5.	10.	10.	-1.	4.	3.
131	AGT018+1	FAIL.	1606.	NA.	673.	0.						
84	AGT024+2	M.	1606.	NA.	1003.	0.	5.	10.	10.	52.	93.	26.
25	ALG074+1	79.	87.	63f	24.	0.	5.	6.	20.	3.	4.	73.
28	ALG211+1	M.	53.	NA.	10.	0.	3.	6.	12.	5.	5.	10.
103	CSR001+1	FAIL.	1030.	NA.	201.	0.						
86	CSR002+1	FAIL.	1029.	NA.	201.	0.						
109	CSR005+2	FAIL.	1031.	NA.	202.	0.						
93	CSR007+1	52.	1030.	-167f	201.	0.	8.	10.	11.	110.	128.	80.
111	CSR019+1	16.	1417.	-138l	256.	0.	3.	10.	6.	5.	5.	14.
119	CSR021+1	1.	1417.	0c	256.	0.	5.	10.	10.	-1.	6.	13.
39	CSR022+1	1.	1417.	0c	256.	0.	5.	10.	10.	-1.	6.	13.
81	CSR024+1	FAIL.										
82	GEO085+1	2.	312.	0c	87.	0.	5.	10.	10.	-1.	6.	6.
123	GEO088+1	3.	312.	1c	89.	0.	7.	10.	13.	4.	5.	35.
129	GEO109+1	FAIL.	358.	NA.	104.	-: 9.						
51	GRA002+1	FAIL.	509.	NA.	121.	0.						
142	GRA002+2	FAIL.	539.	NA.	129.	0.						
104	GRA002+4	4.	516.	1c	122.	0.	7.	12.	7.	-1.	5.	9.
72	GRA004+1	16.	505.	7c	122.	7: 9.	3.	12.	6.	-1.	9.	5.
	GRA004+1	6.	505.	2l	122.	6: 9.	10.	12.	6.	-1.	10.	28.
	GRA004+1	5.	505.	2l	122.	5: 9.	8.	12.	10.	-1.	10.	28.
	GRA004+1	20.	505.	9l	122.	2: 9.	7.	12.	6.	-1.	10.	28.
	GRA004+1	19.	505.	9l	122.	1: 9.	6.	12.	10.	-1.	10.	5.
	GRA004+1	13.	505.	6l	122.	4: 9.	8.	12.	13.	-1.	10.	5.
	GRA004+1	12.	505.	5l	122.	3: 9.	7.	12.	10.	-1.	10.	5.
	GRA004+1	27.	505.	13l	122.	9: 9.	6.	12.	13.	-1.	10.	4.
	GRA004+1	8.	505.	3c	122.	8: 9.	4.	12.	10.	-1.	10.	27.
45	GRA006+1	FAIL.	621.	NA.	133.	-:10.						
110	GRA007+1	90.	516.	74f	122.	0.	6.	12.	9.	302.	354.	104.
88	GRA007+2	66.	526.	50l	123.	0.	6.	12.	8.	271.	550.	105.
19	GRA008+1	FAIL.	509.	NA.	123.	0.						
120	GRA009+2	FAIL.	510.	NA.	124.	0.						
139	GRA010+1	5.	519.	2c	124.	0.	8.	12.	11.	-1.	1.	12.
96	GRA011+1	FAIL.	506.	NA.	121.	0.						
108	GRA012+1	FAIL.	507.	NA.	121.	0.						
38	GRP001+6	M.	41.	NA.	9.	0.	3.	12.	9.	5.	5.	13.
2	GRP012+5	M.	46.	NA.	9.	0.	3.	12.	9.	3.	3.	13.
30	LCL414+1	2.	29.	0c	4.	0.	4.	10.	27.	-1.	7.	5.
52	MGT022+1	M.	43.	NA.	9.	-:10.	4.	8.	14.	9.	9.	22.
54	MGT022+2	M.	43.	NA.	9.	-:10.	4.	8.	14.	4.	4.	22.
56	MGT028+1	M.	156.	NA.	13.	0.	6.	6.	10.	5.	5.	44.
44	MGT036+1	M.	91.	NA.	13.	-: 2.	4.	8.	9.	6.	6.	18.
34	MGT036+3	M.	59.	NA.	8.	0.	4.	8.	9.	6.	6.	9.
132	MGT042+1	4.	249.	1c	57.	4:10.	7.	6.	6.	-1.	2.	12.
	MGT042+1	6.	249.	2l	57.	6:10.	10.	6.	6.	-1.	2.	12.
	MGT042+1	10.	249.	4l	57.	a:10.	5.	6.	6.	-1.	2.	12.
	MGT042+1	2.	249.	0c	57.	2:10.	5.	6.	10.	-1.	2.	12.
	MGT042+1	14.	249.	6c	57.	e:10.	10.	6.	6.	-1.	2.	12.
	MGT042+1	12.	249.	5c	57.	c:10.	8.	6.	10.	-1.	2.	12.
	MGT042+1	9.	249.	4l	57.	9:10.	4.	6.	10.	-1.	2.	42.
	MGT042+1	1.	249.	0c	57.	1:10.	5.	6.	10.	-1.	2.	42.
	MGT042+1	10.	249.	4l	57.	b:10.	5.	6.	6.	-1.	2.	41.
	MGT042+1	2.	249.	0c	57.	3:10.	5.	6.	10.	-1.	3.	42.
	MGT042+1	4.	249.	1c	57.	5:10.	7.	6.	6.	-1.	5.	40.
	MGT042+1	13.	249.	6c	57.	d:10.	8.	6.	13.	-1.	6.	41.
	MGT042+1	23.	249.	11l	57.	7:10.	7.	6.	13.	-1.	6.	41.
	MGT042+1	111.	249.	95f	57.	f:10.	7.	6.	13.	-1.	7.	41.
	MGT042+1	23.	249.	11l	57.	8:10.	7.	6.	13.	-1.	12.	12.
	MGT042+1	111.	249.	95f	57.	10:10.	7.	6.	13.	-1.	12.	25.

THM	NAME	FROM	WEAK	PICK	BASE	SPLT	LITS	VARS	TRMS	KNOW	TIME	LENG
99	MGT060+1	9.	107.	4c	44.	0.	4.	6.	14.	-1.	2.	54.
114	MSC010+1	FAIL.										
137	NUM310+1	M.	1533.	NA.	438.	0.	7.	22.	10.	13.	30.	20.
117	NUM327+1	M.	1533.	NA.	438.	0.	7.	22.	10.	7.	7.	9.
147	NUM329+1	M.	1533.	NA.	438.	0.	7.	22.	10.	8.	8.	9.
73	NUM334+1	M.	1533.	NA.	438.	0.	7.	22.	10.	7.	8.	8.
130	NUM398+1	5.	166.	2l	101.	0.	9.	6.	10.	2.	2.	26.
76	NUM400+1	M.	221.	NA.	127.	-: 8.	3.	8.	12.	30.	47.	52.
20	PUZ047+1	M.	173.	NA.	15.	0.	3.	6.	14.	8.	8.	8.
95	SET008+3	15.	86.	7l	38.	0.	4.	6.	14.	6.	6.	8.
50	SET009+3	13.	36.	6c	8.	0.	8.	6.	18.	-1.	6.	14.
59	SET014+3	3.	58.	1c	27.	0.	6.	6.	16.	-1.	6.	9.
11	SET067+1	6.	488.	2c	133.	0.	8.	8.	12.	-1.	5.	12.
98	SET094+1	M.	489.	NA.	136.	0.	3.	8.	16.	1.	2.	9.
80	SET099+1	FAIL.	493.	NA.	135.	0.						
122	SET144+3	56.	89.	40f	35.	0.	4.	6.	17.	4.	8.	46.
146	SET171+3	57.	89.	41f	34.	0.	4.	6.	33.	5.	20.	80.
143	SET200+3	M.	60.	NA.	27.	0.	3.	6.	14.	6.	6.	11.
150	SET577+3	M.	81.	NA.	31.	0.	3.	6.	12.	3.	4.	18.
140	SET579+3	M.	59.	NA.	24.	0.	3.	6.	12.	1.	1.	21.
128	SET582+3	56.	139.	40l	45.	0.	3.	6.	12.	7.	21.	62.
41	SET584+3	M.	58.	NA.	26.	0.	3.	6.	14.	6.	6.	9.
141	SET587+3	M.	82.	NA.	37.	-: 4.	3.	6.	12.	4.	6.	43.
6	SET588+3	M.	36.	NA.	8.	0.	3.	6.	14.	8.	8.	14.
48	SET589+3	1.	54.	0c	12.	0.	5.	6.	14.	-1.	5.	9.
36	SET590+3	1.	32.	0c	7.	0.	5.	6.	10.	-1.	5.	7.
53	SET593+3	4.	68.	1c	31.	2: 2.	6.	6.	8.	-1.	8.	12.
	SET593+3	7.	68.	3c	31.	1: 2.	3.	6.	16.	-1.	8.	12.
97	SET594+3	M.	91.	NA.	35.	0.	3.	6.	18.	10.	10.	12.
133	SET595+3	5.	99.	2c	38.	0.	8.	6.	14.	2.	2.	29.
127	SET603+3	M.	84.	NA.	36.	0.	3.	6.	12.	6.	7.	8.
136	SET623+3	FAIL.	122.	NA.	40.	0.						
75	SET624+3	M.	53.	NA.	27.	-: c.	3.	6.	12.	3.	3.	43.
10	SET631+3	29.	38.	14v	8.	0.	3.	6.	10.	-1.	6.	7.
135	SET655+3	53.	180.	37l	77.	0.	9.	8.	14.	1.	1.	17.
105	SET656+3	1.	392.	0c	119.	0.	5.	10.	11.	-1.	4.	10.
61	SET657+3	1.	321.	0c	114.	0.	5.	10.	10.	-1.	2.	31.
35	SET669+3	M.	322.	NA.	110.	-: 2.	6.	8.	10.	6.	7.	23.
134	SET680+3	M.	310.	NA.	110.	-:10.	6.	10.	10.	9.	10.	28.
21	SET795+4	26.	1540.	12c	240.	0.	6.	10.	7.	18.	22.	22.
15	SET796+4	FAIL.	1719.	NA.	367.	0.						
121	SET798+4	1.	1539.	0v	241.	0.	5.	10.	10.	-1.	7.	7.
145	SET802+4	2.	1529.	0c	240.	2: c.	5.	10.	10.	-1.	10.	1.
	SET802+4	4.	1529.	1c	240.	4: c.	7.	10.	6.	-1.	10.	1.
	SET802+4	3.	1529.	1c	240.	3: c.	6.	10.	13.	-1.	10.	1.
	SET802+4	6.	1529.	2c	240.	6: c.	10.	10.	6.	-1.	10.	2.
	SET802+4	8.	1529.	3l	240.	8: c.	3.	10.	10.	-1.	10.	2.
	SET802+4	4.	1529.	1c	240.	5: c.	7.	10.	6.	-1.	10.	8.
	SET802+4	7.	1529.	3l	240.	7: c.	8.	10.	13.	-1.	10.	5.
	SET802+4	10.	1529.	-135l	240.	a: c.	6.	10.	6.	-1.	11.	1.
	SET802+4	8.	1529.	3l	240.	9: c.	4.	10.	10.	-1.	11.	1.
	SET802+4	1.	1529.	0c	240.	1: c.	5.	10.	10.	-1.	11.	12.
	SET802+4	96.	1529.	80l	240.	c: c.	4.	10.	9.	-1.	20.	11.
	SET802+4	95.	1529.	79l	240.	b: c.	9.	10.	13.	-1.	27.	13.
65	SET805+4	FAIL.	1531.	NA.	238.	0.						
5	SET806+4	FAIL.	1522.	NA.	202.	0.						
23	SET808+4	FAIL.	589.	NA.	143.	0.						
31	SET811+4	104.	590.	88f	143.	0.	4.	10.	9.	24.	46.	26.
74	SET812+4	5.	588.	2c	142.	0.	9.	10.	10.	17.	38.	23.
69	SET814+4	1.	590.	0c	143.	0.	5.	10.	10.	1.	1.	9.
9	SET815+4	FAIL.	591.	NA.	143.	0.						
112	SET816+4	FAIL.	590.	NA.	143.	0.						
68	SET896+1	M.	13.	NA.	7.	-: 4.	3.	4.	10.	6.	6.	4.
64	SET899+1	M.	21.	NA.	7.	0.	3.	6.	10.	6.	6.	3.
26	SET913+1	M.	17.	NA.	6.	0.	3.	4.	10.	5.	5.	2.
46	SET915+1	M.	17.	NA.	6.	0.	3.	4.	10.	5.	5.	2.
92	SET919+1	M.	169.	NA.	43.	-: 2.	3.	8.	16.	3.	6.	40.

THM	NAME	FROM	WEAK	PICK	BASE	SPLT	LITS	VARS	TRMS	KNOW	TIME	LENG
13	SET935+1	M.	141.	NA.	46.	0.	3.	8.	18.	2.	3.	41.
22	SET939+1	M.	14.	NA.	6.	0.	3.	4.	10.	6.	6.	2.
125	SET942+1	1.	104.	0c	34.	0.	5.	8.	14.	-1.	5.	12.
87	SET951+1	132.	271.	116f	63.	0.	9.	12.	24.	30.	41.	50.
42	SET966+1	M.	28.	NA.	8.	0.	3.	6.	14.	6.	6.	5.
91	SET982+1	FAIL.	346.	NA.	55.	0.						
100	SEU003+1	3.	300.	1c	96.	0.	7.	8.	13.	78.	78.	24.
148	SEU009+1	4.	192.	1c	80.	4: c.	7.	6.	6.	-1.	6.	7.
	SEU009+1	6.	192.	2l	80.	6: c.	10.	6.	6.	-1.	6.	8.
	SEU009+1	4.	192.	1c	80.	5: c.	7.	6.	6.	-1.	6.	7.
	SEU009+1	10.	192.	4l	80.	a: c.	5.	6.	6.	-1.	6.	1.
	SEU009+1	9.	192.	4l	80.	9: c.	4.	6.	10.	-1.	6.	1.
	SEU009+1	14.	192.	6l	80.	2: c.	10.	6.	6.	-1.	6.	1.
	SEU009+1	20.	192.	-140v	80.	8: c.	7.	6.	6.	-1.	6.	10.
	SEU009+1	19.	192.	-140v	80.	7: c.	6.	6.	10.	-1.	6.	10.
	SEU009+1	15.	192.	7l	80.	3: c.	8.	6.	10.	-1.	7.	1.
	SEU009+1	10.	192.	4l	80.	b: c.	5.	6.	6.	-1.	8.	11.
	SEU009+1	13.	192.	-137v	80.	1: c.	8.	6.	13.	-1.	9.	17.
	SEU009+1	84.	192.	68f	80.	c: c.	7.	6.	8.	-1.	10.	15.
118	SEU014+1	FAIL.	204.	NA.	84.	0.						
67	SEU075+1	FAIL.	314.	NA.	106.	0.						
17	SEU086+1	FAIL.	399.	NA.	213.	0.						
107	SWC078+1	FAIL.	1230.	NA.	289.	-:10.						
115	SWC231+1	FAIL.	1317.	NA.	295.	0.						
29	SWC239+1	56.	1310.	40c	298.	8:10.	3.	12.	9.	-1.	3.	9.
	SWC239+1	49.	1310.	33l	298.	1:10.	4.	12.	13.	-1.	4.	11.
	SWC239+1	9.	1310.	4l	298.	9:10.	4.	12.	12.	-1.	4.	11.
	SWC239+1	10.	1310.	4l	298.	a:10.	5.	12.	8.	-1.	4.	20.
	SWC239+1	66.	1310.	50l	298.	2:10.	5.	12.	10.	-1.	5.	20.
	SWC239+1	56.	1310.	40c	298.	b:10.	3.	12.	9.	-1.	7.	9.
	SWC239+1	49.	1310.	33l	298.	3:10.	4.	12.	13.	-1.	8.	11.
	SWC239+1	56.	1310.	40c	298.	c:10.	3.	12.	9.	-1.	9.	9.
	SWC239+1	56.	1310.	40c	298.	d:10.	3.	12.	9.	-1.	12.	9.
	SWC239+1	49.	1310.	33l	298.	4:10.	4.	12.	13.	-1.	12.	11.
	SWC239+1	56.	1310.	40c	298.	e:10.	3.	12.	9.	-1.	15.	9.
	SWC239+1	49.	1310.	33l	298.	5:10.	4.	12.	13.	-1.	18.	11.
	SWC239+1	56.	1310.	40c	298.	f:10.	3.	12.	9.	-1.	19.	9.
	SWC239+1	54.	1310.	38c	298.	6:10.	10.	12.	9.	-1.	19.	20.
	SWC239+1	49.	1310.	33l	298.	7:10.	4.	12.	13.	-1.	28.	11.
	SWC239+1	72.	1310.	56l	298.	10:10.	4.	12.	14.	-1.	39.	25.
66	SWV011+1	M.	97.	NA.	16.	0.	3.	2.	35.	5.	5.	2.
18	SWV014+1	6.	216.	2l	37.	0.	9.	14.	14.	1.	1.	28.
113	SWV026+1	FAIL.	2000.	NA.	467.	-: 7.						
55	SWV030+1	6.	913.	2v	198.	6:10.	11.	24.	6.	-1.	5.	14.
	SWV030+1	26.	913.	12l	198.	a:10.	6.	24.	6.	-1.	5.	10.
	SWV030+1	4.	913.	1l	198.	4:10.	8.	24.	6.	-1.	5.	14.
	SWV030+1	41.	913.	25l	198.	9:10.	4.	24.	11.	-1.	6.	3.
	SWV030+1	30.	913.	14l	198.	e:10.	10.	24.	7.	-1.	6.	16.
	SWV030+1	2.	913.	-131l	198.	2:10.	5.	24.	10.	-1.	7.	3.
	SWV030+1	49.	913.	33l	198.	1:10.	4.	24.	11.	-1.	7.	3.
	SWV030+1	4.	913.	1l	198.	5:10.	8.	24.	6.	-1.	8.	13.
	SWV030+1	26.	913.	12l	198.	b:10.	6.	24.	6.	-1.	9.	17.
	SWV030+1	6.	913.	2v	198.	7:10.	11.	24.	6.	-1.	11.	21.
	SWV030+1	76.	913.	60v	198.	c:10.	8.	24.	8.	-1.	12.	14.
	SWV030+1	30.	913.	14l	198.	f:10.	11.	24.	7.	-1.	13.	24.
	SWV030+1	34.	913.	18c	198.	3:10.	6.	24.	7.	-1.	14.	17.
	SWV030+1	6.	913.	2v	198.	8:10.	11.	24.	6.	-1.	15.	16.
	SWV030+1	76.	913.	60v	198.	d:10.	7.	24.	8.	-1.	15.	17.
	SWV030+1	26.	913.	12l	198.	10:10.	6.	24.	6.	-1.	61.	16.
106	SWV033+1	FAIL.	880.	NA.	177.	-:10.						
43	SWV036+1	FAIL.										
77	SWV038+1	FAIL.										
47	SWV039+1	FAIL.	2000.	NA.	442.	-: 7.						
149	SWV041+1	M.	809.	NA.	179.	0.	11.	24.	10.	2.	3.	8.
63	SWV048+1	114.	805.	98f	172.	0.	6.	24.	9.	6.	7.	19.
33	SWV050+1	13.	821.	-137f	163.	0.	9.	24.	13.	159.	159.	19.
101	SWV053+1	FAIL.	1134.	NA.	184.	-: 9.						
90	SWV090+1	FAIL.	814.	NA.	173.	-: 5.						

THM	NAME	FROM	WEAK	PICK	BASE	SPLT	LITS	VARS	TRMS	KNOW	TIME	LENG
71	SWV092+1	4.	872.	1c	189.	2: 2.	7.	24.	6.	-1.	1.	8.
	SWV092+1	4.	872.	1c	189.	1: 2.	7.	24.	6.	-1.	2.	8.
37	SWV094+1	FAIL.	1251.	NA.	244.	-: c.						
94	SWV096+1	10.	976.	4c	212.	2: 2.	5.	24.	6.	-1.	2.	5.
	SWV096+1	10.	976.	4c	212.	1: 2.	5.	24.	6.	-1.	2.	5.
57	SWV098+1	24.	962.	11l	202.	2: 2.	4.	24.	6.	-1.	2.	12.
	SWV098+1	24.	962.	11l	202.	1: 2.	4.	24.	6.	-1.	8.	12.
85	SWV103+1	FAIL.	778.	NA.	164.	-: 5.						
89	SWV117+1	FAIL.	2000.	NA.	805.	-:10.						
144	SWV161+1	64.	802.	48l	171.	0.	3.	24.	8.	4.	5.	11.
78	SWV173+1	10.	864.	4l	178.	0.	5.	24.	6.	-1.	1.	3.
7	SWV177+1	10.	870.	4l	180.	0.	5.	24.	6.	-1.	2.	3.
126	SWV200+1	M.	928.	NA.	204.	-: 8.	11.	24.	10.	6.	6.	5.
3	SWV209+1	20.	914.	9l	196.	0.	7.	24.	6.	2.	3.	26.
83	SWV233+1	FAIL.	171.	NA.	60.	0.						
124	SWV234+1	FAIL.	165.	NA.	35.	0.						
1	SWV234+2	FAIL.	106.	NA.	31.	0.						
49	SWV235+1	10.	182.	-135f	38.	0.	5.	12.	12.	-1.	1.	25.
27	SWV236+1	24.	177.	-142f	37.	0.	6.	12.	12.	2.	3.	10.
79	SWV237+1	M.	130.	NA.	31.	-: c.	3.	6.	16.	5.	5.	9.
58	SWV938+1	FAIL.										
70	SYN081+1	M.	11.	NA.	2.	0.	3.	2.	10.	6.	6.	3.
4	SYN084+1	M.	266.	NA.	30.	-:10.	4.	4.	9.	1.	2.	32.
14	SYN364+1	M.	17.	NA.	4.	-: 4.	3.	6.	10.	6.	6.	9.
16	SYN365+1	3.	24.	1c	5.	0.	6.	3.	13.	-1.	7.	6.
12	SYN728+1	M.	17.	NA.	4.	-: 4.	3.	6.	10.	5.	5.	9.
62	SYN730+1	M.	7.	NA.	1.	0.	3.	4.	14.	7.	7.	1.
60	SYN939+1	M.	10.	NA.	4.	-: c.	3.	4.	9.	8.	8.	11.
8	SYN941+1	M.	2.	NA.	3.	-: c.	3.	4.	9.	7.	7.	7.
32	SYN940+1	M.	1.	NA.	3.	-: c.	3.	4.	9.	8.	8.	7.
24	SYN943+1	M.	25.	NA.	6.	0.	3.	2.	10.	5.	5.	8.
40	SYN980+1	M.	9.	NA.	3.	-: 4.	3.	4.	10.	7.	7.	7.

APPENDIX 5: OCTOPUS MIX RESULTS CASC 2006

THM	NAME	FROM	WEAK	PICK	BASE	SPLT	LITS	VARS	TRMS	KNOW	TIME	LENG
64	ANA002-1	23.	101.	11l	17.	0.	5.	6.	12.	3.	7.	69.
57	ANA003-2	63.	131.	47f	16.	0.	4.	6.	44.	4.	10.	25.
34	ANA004-5	54.	114.	38f	15.	0.	8.	6.	22.	4.	4.	30.
71	ANA008-2	4.	24.	1c	19.	0.	6.	8.	18.	-1.	7.	6.
129	ANA014-2	FAIL.	71.	NA.	33.	0.						
68	ANA016-2	39.	43.	23f	21.	0.	7.	13.	28.	1.	1.	19.
96	ANA017-2	42.	77.	26f	27.	0.	4.	14.	25.	3.	3.	13.
124	ANA022-1	FAIL.										
149	ANA037-1	FAIL.										
61	ANA030-2	FAIL.	118.	NA.	46.	0.						
31	ANA040-2	66.	24.	2c	19.	0.	5.	8.	20.	-1.	3.	10.
51	ANA044-2	6.	47.	2c	26.	0.	8.	8.	17.	-1.	5.	12.
66	ANA045-2	3.	22.	1c	19.	0.	6.	8.	32.	-1.	4.	8.
35	BOO020-1	FAIL.	39.	NA.	9.	-: 3.						
1	COL003-10	FAIL.	43.	NA.	9.	0.						
21	COL003-2	107.	41.	9v	9.	0.	5.	6.	39.	51.	93.	25.
86	COL042-3	FAIL.	43.	NA.	9.	0.						
142	COL093-2	119.	1186.	103l	385.	3: 4.	2.	18.	22.	-1.	12.	2.
	COL093-2	128.	1186.	112l	385.	4: 4.	3.	18.	23.	-1.	12.	2.
	COL093-2	122.	1186.	106l	385.	2: 4.	5.	18.	23.	-1.	13.	2.
	COL093-2	2.	1186.	0c	385.	1: 4.	5.	18.	19.	-1.	114.	27.
147	COL112-1	M.	2000.	NA.	1857.	0.	4.	32.	24.	5.	5.	2.
119	COL119-1	M.	2000.	NA.	1865.	0.	4.	32.	24.	5.	5.	2.
29	COM005-1	FAIL.	2000.	NA.	267.	0.						
128	COM009-1	M.	2000.	NA.	1798.	-: 4.	3.	32.	40.	23.	23.	6.
94	FLD013-1	10.	163.	4c	33.	0.	6.	10.	12.	4.	4.	14.
44	FLD060-3	3.	147.	1c	29.	0.	6.	14.	18.	1.	1.	15.
98	GEO006-1	4.	191.	1c	59.	2: 2.	7.	16.	6.	-1.	1.	4.
	GEO006-1	18.	191.	8l	59.	1: 2.	6.	16.	10.	-1.	4.	27.
106	GEO007-1	18.	200.	-139f	59.	0.	6.	16.	10.	11.	19.	74.
105	GEO011-1	89.	213.	73f	62.	0.	5.	16.	12.	2.	4.	32.
33	GEO051-3	11.	585.	5l	121.	0.	6.	16.	10.	-1.	1.	9.
114	GEO076-4	94.	297.	78v	59.	0.	11.	10.	9.	8.	12.	77.
122	GEO083-1	65.	318.	49l	89.	0.	4.	10.	12.	-1.	1.	12.
117	GEO089-1	110.	310.	94f	88.	0.	11.	10.	9.	36.	37.	32.
132	GRP039-5	14.	52.	6v	22.	0.	4.	6.	10.	18.	45.	114.
5	GRP070-1	FAIL.	16.	NA.	10.	-: 3.						
75	GRP061-1	FAIL.	13.	NA.	7.	-: 3.						
70	GRP074-1	FAIL.	16.	NA.	10.	-: 3.						
90	GRP082-1	FAIL.	18.	NA.	10.	-: 3.						
15	GRP085-1	FAIL.	8.	NA.	7.	-: 4.						
80	GRP086-1	FAIL.	8.	NA.	7.	-: 4.						
30	GRP102-1	FAIL.	27.	NA.	12.	-: 4.						
60	GRP103-1	FAIL.	27.	NA.	12.	-: 4.						
25	GRP110-1	FAIL.	15.	NA.	10.	-: 4.						
50	GRP311-1	5.	283.	2c	39.	5: e.	8.	8.	15.	-1.	1.	1.
	GRP311-1	4.	283.	1l	39.	4: e.	7.	8.	9.	-1.	3.	31.
	GRP311-1	56.	283.	40l	39.	e: e.	3.	8.	10.	-1.	3.	51.
	GRP311-1	36.	283.	20v	39.	8: e.	7.	8.	10.	-1.	6.	49.
	GRP311-1	16.	283.	7l	39.	2: e.	4.	8.	9.	-1.	7.	149.
	GRP311-1	56.	283.	40l	39.	1: e.	3.	8.	10.	-1.	8.	47.
	GRP311-1	16.	283.	7l	39.	3: e.	3.	8.	9.	-1.	10.	61.
	GRP311-1	21.	283.	10l	39.	7: e.	8.	8.	15.	-1.	15.	49.
	GRP311-1	16.	283.	7l	39.	6: e.	4.	8.	9.	-1.	27.	293.
	GRP311-1	24.	283.	11l	39.	a: e.	4.	8.	9.	-1.	28.	163.
	GRP311-1	95.	283.	79c	39.	b: e.	13.	8.	19.	-1.	62.	202.
	GRP311-1	26.	283.	12l	39.	c: e.	6.	8.	9.	-1.	73.	185.
	GRP311-1	26.	283.	12l	39.	d: e.	6.	8.	9.	-1.	84.	78.
	GRP311-1	104.	283.	88l	39.	9: e.	4.	8.	13.	-1.	249.	265.
26	HWV002-1	FAIL.	192.	NA.	61.	-: 3.						

THM	NAME	FROM	WEAK	PICK	BASE	SPLT	LITS	VARS	TRMS	KNOW	TIME	LENG
45	LAT198-1	FAIL.	82.	NA.	20.	0.						
20	HWV003-2	FAIL.	182.	NA.	45.	-: 2.						
145	HWV016-1	21.	804.	10c	124.	0.	9.	8.	10.	4.	7.	89.
78	LAT003-1	FAIL.	118.	NA.	37.	0.						
10	LAT005-4	FAIL.	98.	NA.	26.	0.						
55	LAT201-1	FAIL.	80.	NA.	20.	0.						
95	LAT209-1	FAIL.	81.	NA.	20.	0.						
85	LAT219-1	FAIL.	82.	NA.	21.	0.						
76	LAT263-2	16.	68.	7c	30.	0.	4.	8.	12.	-1.	3.	14.
59	LAT273-2	M.	87.	NA.	6.	0.	3.	12.	18.	6.	6.	8.
46	LAT277-2	M.	52.	NA.	29.	0.	3.	10.	22.	5.	5.	6.
92	LCL028-1	FAIL.	42.	NA.	4.	0.						
81	LCL109-4	FAIL.	57.	NA.	20.	0.						
87	LCL124-1	FAIL.	38.	NA.	2.	0.						
72	LCL125-1	FAIL.	35.	NA.	3.	0.						
56	LCL145-1	FAIL.	58.	NA.	20.	0.						
91	LCL151-1	FAIL.	61.	NA.	20.	0.						
41	LCL185-3	50.	44.	34f	18.	0.	4.	6.	12.	3.	3.	13.
6	LCL222-3	FAIL.	52.	NA.	18.	0.						
62	LCL230-1	81.	68.	65f	8.	0.	7.	6.	34.	1.	1.	17.
36	LCL237-3	35.	53.	19f	21.	0.	4.	6.	21.	6.	6.	21.
27	LCL249-1	FAIL.	67.	NA.	8.	0.						
2	LCL253-1	FAIL.	72.	NA.	8.	0.						
17	LCL423-1	FAIL.	21.	NA.	2.	-: 3.						
120	LCL434-2	54.	43.	38f	20.	0.	7.	8.	13.	-1.	4.	8.
150	LCL439-1	84.	2000.	68l	1795.	0.	2.	34.	28.	-1.	30.	6.
24	LCL442-2	M.	140.	NA.	11.	-: c.	3.	8.	48.	7.	7.	16.
52	LCL444-2	FAIL.	84.	NA.	6.	0.						
115	MGT034-2	42.	304.	26l	61.	0.	6.	10.	11.	1.	1.	48.
84	NLP080-1	FAIL.	0.	NA.	50.	0.	0.	0.	0.	-1.	1.	-1.
102	NUM010-1	60.	1709.	44l	406.	0.	8.	29.	8.	18.	37.	51.
12	NUM017-1	64.	65.	49f	23.	0.	4.	12.	8.	1.	1.	22.
109	NUM058-1	FAIL.	1127.	NA.	265.	0.						
7	PLA014-1	11.	159.	-136f	30.	0.	7.	10.	18.	2.	2.	39.
37	PLA010-1	6.	159.	2c	30.	0.	9.	10.	12.	47.	47.	42.
97	PLA016-1	1.	155.	0c	30.	0.	4.	10.	14.	1.	1.	22.
77	PLA018-1	43.	159.	27v	30.	0.	4.	10.	25.	46.	75.	60.
19	PUZ035-4	1.	49.	0c	13.	1: 6.	5.	6.	14.	-1.	8.	1.
	PUZ035-4	2.	49.	0c	13.	2: 6.	4.	6.	14.	-1.	8.	1.
	PUZ035-4	4.	49.	34v	13.	4: 6.	6.	6.	9.	-1.	8.	3.
	PUZ035-4	6.	49.	35f	13.	6: 6.	8.	6.	9.	-1.	8.	3.
	PUZ035-4	11.	49.	5l	13.	5: 6.	6.	6.	14.	-1.	8.	3.
	PUZ035-4	15.	49.	7l	13.	3: 6.	3.	6.	14.	-1.	8.	3.
67	RNG004-3	10.	106.	4l	33.	0.	6.	14.	8.	1.	1.	18.
100	RNG027-2	FAIL.	140.	NA.	32.	0.						
40	RNG029-3	FAIL.	162.	NA.	34.	0.						
69	SET013-1	14.	68.	1c	21.	0.	4.	6.	9.	3.	3.	19.
135	SET016-3	FAIL.	899.	NA.	269.	0.						
110	SET018-3	72.	916.	-187f	273.	0.	4.	29.	12.	63.	95.	19.
134	SET018-4	FAIL.	901.	NA.	271.	0.						
108	SET022-3	11.	935.	5l	273.	0.	6.	28.	10.	-1.	2.	10.
138	SET027-3	64.	951.	48l	279.	0.	3.	29.	8.	-1.	1.	7.
127	SET028-3	123.	957.	107l	278.	0.	7.	31.	17.	15.	22.	29.
107	SET035-3	FAIL.	992.	NA.	287.	0.						
139	SET208-6	4.	797.	1c	188.	0.	7.	20.	16.	17.	19.	23.
133	SET787-2	2.	793.	0c	276.	0.	4.	18.	20.	-1.	16.	13.
148	SET820-2	3.	17.	1c	16.	0.	6.	8.	18.	-1.	7.	8.
104	SET823-1	7.	2000.	-134l	1779.	0.	2.	32.	53.	-1.	3.	7.
23	SET827-1	M.	2000.	NA.	1776.	0.	3.	32.	40.	5.	5.	3.
99	SET830-2	1.	68.	0c	9.	0.	4.	8.	12.	1.	1.	17.

THM	NAME	FROM	WEAK	PICK	BASE	SPLT	LITS	VARS	TRMS	KNOW	TIME	LENG
103	SET831-2	29.	115.	14l	28.	d:10.	3.	8.	14.	-1.	6.	2.
	SET831-2	30.	115.	14l	28.	e:10.	4.	8.	10.	-1.	6.	7.
	SET831-2	28.	115.	13l	28.	c:10.	7.	8.	14.	-1.	6.	7.
	SET831-2	32.	115.	15v	28.	10:10.	6.	8.	15.	-1.	6.	7.
	SET831-2	34.	115.	18c	28.	2:10.	8.	8.	10.	-1.	6.	7.
	SET831-2	33.	115.	17l	28.	1:10.	4.	8.	19.	-1.	6.	2.
	SET831-2	6.	115.	2l	28.	6:10.	8.	8.	9.	-1.	6.	2.
	SET831-2	31.	115.	15v	28.	f:10.	5.	8.	15.	-1.	6.	7.
	SET831-2	36.	115.	20c	28.	4:10.	3.	8.	10.	-1.	6.	7.
	SET831-2	40.	115.	24c	28.	8:10.	7.	8.	10.	-1.	6.	2.
	SET831-2	35.	115.	19l	28.	3:10.	3.	8.	15.	-1.	6.	7.
	SET831-2	42.	115.	26c	28.	a:10.	3.	8.	15.	-1.	6.	7.
	SET831-2	104.	115.	88f	28.	9:10.	9.	8.	12.	-1.	39.	28.
	SET831-2	117.	115.	101f	28.	5:10.	8.	8.	21.	-1.	52.	36.
	SET831-2	86.	115.	70v	28.	7:10.	5.	8.	11.	-1.	61.	25.
	SET831-2	27.	115.	13l	28.	b:10.	9.	8.	18.	-1.	92.	81.
101	SET834-1	FAIL.	2000.	NA.	1778.	0.						
14	SET834-2	5.	68.	2c	9.	0.	8.	8.	12.	1.	1.	17.
65	SET838-2	M.	30.	NA.	9.	0.	3.	6.	21.	3.	4.	14.
125	SET841-1	M.	2000.	NA.	1783.	0.	4.	32.	24.	5.	7.	17.
58	SET853-1	26.	2000.	12c	1799.	0.	7.	32.	16.	-1.	22.	45.
73	SET861-2	FAIL.	224.	NA.	47.	0.						
4	SET863-2	2.	181.	0c	13.	0.	5.	12.	18.	1.	1.	24.
18	SWC169-1	FAIL.	1068.	NA.	287.	-:10.						
126	SWC225-1	FAIL.	1170.	NA.	291.	-: 4.						
42	SWV014-1	57.	195.	41v	33.	0.	3.	14.	30.	1.	1.	26.
140	SWV239-1	FAIL.	2000.	NA.	1829.	0.						
116	SWV242-1	M.	2000.	NA.	1820.	-: 2.	3.	32.	40.	11.	12.	10.
53	SWV243-1	FAIL										
112	SWV247-1	FAIL.	2000.	NA.	1834.	0.						
39	SWV248-2	34.	79.	18l	8.	0.	9.	8.	22.	1.	1.	20.
11	SWV249-2	FAIL.	150.	NA.	38.	0.						
16	SWV251-2	M.	106.	NA.	33.	0.	3.	8.	25.	7.	8.	14.
3	SWV254-1	FAIL.										
13	SWV264-1	7.	2000.	3l	1807.	0.	2.	32.	53.	-1.	5.	2.
130	SWV256-1	FAIL.										
63	SWV267-1	FAIL.										
28	SWV270-1	FAIL.										
146	SWV273-1	FAIL.										
123	SWV274-2	1.	48.	35f	24.	0.	4.	8.	16.	3.	4.	18.
144	SWV275-2	8.	59.	3c	25.	0.	3.	8.	24.	2.	2.	14.
111	SWV281-2	4.	37.	1c	23.	0.	6.	8.	9.	-1.	5.	10.
118	SWV282-2	102.	98.	86f	41.	0.	11.	8.	13.	3.	3.	28.
54	SWV291-2	3.	135.	1c	10.	0.	6.	8.	64.	-1.	7.	10.
83	SWV302-1	7.	2000.	3l	1943.	0.	2.	32.	32.	-1.	5.	5.
89	SWV305-2	M.	115.	NA.	10.	-: 2.	3.	8.	36.	6.	7.	17.
8	SWV315-1	FAIL.										
141	SWV317-1	FAIL.										
93	SWV319-2	23.	76.	11l	30.	0.	5.	8.	48.	1.	2.	10.
79	SWV323-2	M.	145.	NA.	16.	-: 8.	3.	26.	36.	4.	5.	29.
113	SWV325-2	3.	112.	1c	36.	1: 2.	6.	8.	42.	-1.	7.	11.
	SWV325-2	2.	112.	0c	36.	2: 2.	4.	8.	32.	-1.	7.	11.
9	SWV329-2	M.	106.	NA.	9.	0.	3.	8.	32.	6.	6.	9.
136	SWV341-1	FAIL.										
48	SWV349-1	FAIL.										
137	SWV342-1	FAIL.										
38	SWV343-1	FAIL.										
121	SWV354-1	FAIL.										
143	SWV354-2	FAIL.	202.	NA.	38.	-: 3.						
43	SWV355-1	FAIL.										
88	SYN014-1	33.	147.	17l	23.	1: 2.	5.	6.	14.	-1.	2.	43.
	SYN014-1	64.	147.	48c	23.	2: 2.	4.	6.	8.	-1.	5.	81.
131	SYN015-1	FAIL.	147.	NA.	22.	0.						

THM	NAME	FROM	WEAK	PICK	BASE	SPLT	LITS	VARS	TRMS	KNOW	TIME	LENG
49	SYN036-3	91.	119.	75l	35.	7: c.	3.	4.	12.	-1.	4.	1.
	SYN036-3	87.	119.	71l	35.	3: c.	5.	4.	14.	-1.	5.	5.
	SYN036-3	78.	119.	62l	35.	6: c.	10.	4.	11.	-1.	7.	24.
	SYN036-3	94.	119.	78l	35.	a: c.	5.	4.	9.	-1.	8.	33.
	SYN036-3	81.	119.	65l	35.	9: c.	6.	4.	11.	-1.	8.	5.
	SYN036-3	108.	119.	92l	35.	c: c.	7.	4.	12.	-1.	9.	26.
	SYN036-3	65.	119.	49l	35.	5: c.	4.	5.	11.	-1.	11.	69.
	SYN036-3	121.	119.	105f	35.	1: c.	4.	5.	13.	-1.	13.	13.
	SYN036-3	98.	119.	82l	35.	2: c.	4.	4.	12.	-1.	34.	48.
	SYN036-3	116.	119.	100l	35.	8: c.	6.	6.	9.	-1.	40.	63.
	SYN036-3	95.	119.	79l	35.	b: c.	7.	4.	12.	-1.	47.	55.
	SYN036-3	123.	119.	107v	35.	4: c.	6.	4.	16.	-1.	52.	49.
74	SYN067-3	FAIL.	500.	NA.	45.	-: 6.						
22	SYN600-1	3.	263.	1c	28.	0.	6.	8.	53.	1.	1.	16.
47	SYN640-1	8.	452.	3c	41.	0.	3.	8.	24.	5.	5.	13.
32	SYN707-1	3.	1003.	1v	82.	0.	6.	8.	53.	-1.	6.	14.
82	SYN708-1	4.	1003.	1v	82.	0.	6.	8.	26.	-1.	1.	14.

APPENDIX 6: OCTOPUS FOF & MIX RESULTS NO LEARNING STRATEGY

On September 4, 2006, OctopusNL was tested on the 31 theorems that Theo failed to prove but that Octopus did. OctopusNL failed to prove 16 of them. Data from the 15 that it did are presented here.

THM NAME	FROM	BASE	SPLT	LITS	VARS	TRMS	KNOW	TIME	LENG
AGT007+2	36.	1003.	0.	7.	10.	7.	-1.	269.	43.
ALG074+1	94.	24.	0.	5.	6.	15.	-1.	47.	71.
NUM310+1	66.	438.	0.	5.	22.	8.	-1.	12.	13.
SET582+3	50.	45.	0.	3.	6.	12.	-1.	47.	80.
SET802+4	10.	240.	a: c.	5.	10.	6.	-1.	9.	1.
SET802+4	2.	240.	2: c.	5.	10.	10.	-1.	10.	1.
SET802+4	4.	240.	4: c.	7.	10.	6.	-1.	10.	1.
SET802+4	5.	240.	5: c.	8.	10.	10.	-1.	10.	8.
SET802+4	21.	240.	9: c.	8.	10.	10.	-1.	10.	1.
SET802+4	18.	240.	6: c.	5.	10.	10.	-1.	10.	2.
SET802+4	3.	240.	3: c.	6.	10.	13.	-1.	10.	1.
SET802+4	8.	240.	8: c.	3.	10.	10.	-1.	10.	2.
SET802+4	7.	240.	7: c.	8.	10.	13.	-1.	10.	8.
SET802+4	13.	240.	1: c.	8.	10.	13.	-1.	11.	12.
SET802+4	84.	240.	c: c.	7.	10.	8.	-1.	23.	16.
SET802+4	80.	240.	b: c.	3.	10.	8.	-1.	41.	12.
SET811+4	56.	143.	0.	3.	10.	7.	-1.	275.	29.
SET812+4	56.	142.	0.	3.	10.	7.	-1.	73.	19.
SEU003+1	108.	96.	0.	7.	8.	13.	-1.	302.	20.
PLA016-1	76.	30.	0.	8.	10.	11.	-1.	3.	22.
SET028-3	123.	278.	0.	6.	31.	17.	-1.	56.	25.
SET831-2	13.	28.	d:10.	8.	8.	18.	-1.	8.	2.
SET831-2	10.	28.	a:10.	5.	8.	9.	-1.	8.	7.
SET831-2	6.	28.	6:10.	8.	8.	9.	-1.	8.	2.
SET831-2	18.	28.	2:10.	5.	8.	14.	-1.	8.	7.
SET831-2	1.	28.	1:10.	5.	8.	14.	-1.	8.	2.
SET831-2	13.	28.	e:10.	8.	8.	18.	-1.	8.	7.
SET831-2	8.	28.	8:10.	3.	8.	14.	-1.	8.	2.
SET831-2	12.	28.	c:10.	7.	8.	14.	-1.	8.	7.
SET831-2	3.	28.	3:10.	6.	8.	18.	-1.	8.	7.
SET831-2	4.	28.	4:10.	6.	8.	9.	-1.	8.	7.
SET831-2	63.	28.	f:10.	7.	8.	20.	-1.	8.	7.
SET831-2	16.	28.	10:10.	4.	8.	9.	-1.	9.	7.
SET831-2	100.	28.	5:10.	4.	8.	12.	-1.	39.	24.
SET831-2	86.	28.	7:10.	4.	8.	11.	-1.	58.	24.
SET831-2	121.	28.	9:10.	4.	8.	18.	-1.	59.	24.
SET831-2	100.	28.	b:10.	4.	8.	12.	-1.	335.	95.
SET853-1	26.	1799.	0.	7.	32.	16.	-1.	12.	45.
SWV248-2	50.	8.	0.	3.	8.	22.	-1.	66.	56.
SWV282-2	100.	41.	0.	7.	8.	9.	-1.	191.	43.
SYN014-1	5.	23.	1: 2.	8.	6.	10.	-1.	3.	42.
SYN014-1	58.	23.	2: 2.	5.	6.	11.	-1.	5.	131.
SYN036-3	10.	35.	a: c.	5.	4.	6.	-1.	19.	33.
SYN036-3	13.	35.	1: c.	8.	4.	12.	-1.	20.	13.
SYN036-3	21.	35.	9: c.	3.	4.	9.	-1.	20.	5.
SYN036-3	18.	35.	6: c.	5.	4.	9.	-1.	20.	24.
SYN036-3	15.	35.	3: c.	3.	4.	9.	-1.	20.	5.
SYN036-3	7.	35.	7: c.	3.	4.	12.	-1.	20.	1.
SYN036-3	2.	35.	2: c.	4.	4.	9.	-1.	21.	46.
SYN036-3	12.	5.	c: c.	7.	4.	9.	-1.	21.	26.
SYN036-3	10.	35.	b: c.	5.	4.	6.	-1.	21.	44.
SYN036-3	4.	35.	4: c.	6.	4.	6.	-1.	21.	49.
SYN036-3	8.	35.	8: c.	3.	6.	9.	-1.	22.	67.
SYN036-3	29.	35.	5: c.	3.	6.	9.	-1.	23.	46.