

ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, IL 60439

ANL/MCS-TM-172

**An Entry in the 1992 Overbeek
Theorem-Proving Contest**

by

E. L. Lusk and W. W. McCune

Mathematics and Computer Science Division

Technical Memorandum No. 172

November 1992

This work was supported by the Office of Scientific Computing, U.S. Department of Energy, under Contract W-31-109-Eng-38.

Contents

Abstract	1
1 Introduction	1
2 Results	1
3 Settings and Set of Support	2
3.1 Settings for the Basic Set	2
3.2 Settings for the Equality Set	2
3.3 Description of the Settings	2
4 Failures on Equality Theorems 6–10	5
5 Summary of OTTER Outputs for the Basic Set	6
5.1 Theorem 1: $x^2 = e$ Groups are Commutative (P-form)	6
5.2 Theorem 2: The Commutator Theorem (P-form)	8
5.3 Theorem 3: $x^2 = x$ Rings are Commutative (P-form)	11
5.4 Theorem 4: Equivalential Calculus, XGK → PYO	14
5.5 Theorem 5: Implicational Calculus Single Axiom, CD-67 (Imp-4)	16
5.6 Theorem 6: Many-valued Sentential Calculus, CD-57	19
5.7 Theorem 7: Many-valued Sentential Calculus, CD-60	21
6 Summary of OTTER Outputs for the Equality Set	23
6.1 Theorem EQ-1: The Commutator Theorem	23
6.2 Theorem EQ-2: Robbins Algebra, $(\exists c, c + c = c) \rightarrow$ Boolean	25
6.3 Theorem EQ-3: On Ternary Boolean Algebra	28
6.4 Theorem EQ-4: Group Theory Single Axiom	31
6.5 Theorem EQ-5: On Wajsberg Algebra	35
7 Conclusion	38
References	39

An Entry in the 1992 Overbeek Theorem-Proving Contest

by

Ewing L. Lusk and William W. McCune

Abstract

At CADE-10 Ross Overbeek proposed a contest to stimulate and reward work in automated theorem proving. This paper represents an entry, or perhaps a family of related entries, in the contest.

1 Introduction

The Conference on Automated Deduction (CADE) has been for nearly twenty years a meeting where both theoreticians and system implementors present their work. Feeling perhaps that the conference was becoming dominated by the theoreticians, Ross Overbeek proposed at CADE-10 in 1990 a contest to stimulate work on the implementation and use of theorem-proving systems. The challenge was to prove a set of theorems, and do so with a uniform approach. That is, it was not allowed to set parameters in the system to specialize it for individual problems. There were actually two separate contests, one represented by a set of seven problems designed to test basic inference components, and the other represented by a set of ten problems designed to test equality-based systems.

This paper describes our experiences in preparing to enter the contest with OTTER [5, 6] and ROO [1, 2], two systems developed at Argonne National Laboratory. ROO is a parallel version of OTTER, but has such different behavior in some cases that we treat them as separate entries. We entered each of them in both contests.

Some of the problems are difficult ones; and although many of the problems had been done before with OTTER, in each case we had set OTTER's many input parameters in a way customized to the problem at hand, and chosen a set of support that appeared to us to be most natural. It was a challenge to come up with a uniform set of parameter settings and a uniform algorithm for picking the set of support that would allow OTTER to prove each of the theorems.

2 Results

OTTER and ROO proved all seven theorems in the basic set first five of the ten problems in the equality set. See Section 3 for the options settings and set of support used.

Tables 1 and 2 list the results on the two sets for OTTER, for ROO running with 8 processors, and for ROO with 12 processors.

The OTTER jobs were run on SPARCstation 2. We used OTTER 2.2, the version that was released in July 1991. The Roo jobs were run on an Alliant 2800 with 12 (Intel i860) processors. The version of Roo we used is based on OTTER 2.2xa+ (June 1992).

3 Settings and Set of Support

Within each set, all of the OTTER jobs used the same settings. However, the settings for the basic set were substantially different from those for the equality set. The Roo jobs used settings slightly different from the OTTER jobs, and (for small technical reasons) the Roo settings for the basic set varied slightly, depending on whether equality is present.

For the basic set, the initial set of support consisted of the positive input clauses, except ($x=x$). For the equality set, the initial set of support depended on whether the theorem has an obvious special hypothesis—if so, then the set of support was the special hypothesis and the denial of the conclusion; if not, the set of support consisted of all input clauses.

The rules for the equality set state that an ordering on the symbols may be included with the input clauses. The ordering is used to orient equality literals.

3.1 Settings for the Basic Set

OTTER: basic set	Roo: basic with equality	Roo: basic without equality
	set(index_for_back_demod)	
set(hyper_res)	set(hyper_res)	set(hyper_res)
set(back_demod)	set(back_demod)	
set(dynamic_demod_all)	set(dynamic_demod_all)	
assign(pick_given_ratio,5)	assign(pick_given_ratio,5)	assign(pick_given_ratio,5)
clear(print_kept)	clear(print_kept)	clear(print_kept)
assign(max_mem,20000)	assign(max_mem,32000)	assign(max_mem,32000)
set(control_memory)	set(control_memory)	set(control_memory)

3.2 Settings for the Equality Set

OTTER: equality set	Roo: equality set
set(knuth_bendix)	set(knuth_bendix)
set(index_for_back_demod)	set(index_for_back_demod)
set(process_input)	set(process_input)
assign(max_mem,16000)	assign(max_mem,32000)
set(control_memory)	set(control_memory)
set(lex_rpo)	set(lex_rpo)
clear(print_kept)	clear(print_kept)
clear(print_new_demod)	clear(print_new_demod)
clear(print_back_demod)	clear(print_back_demod)

3.3 Description of the Settings

set(hyper_res). This option activates the inference rule hyperresolution.

set(back_demod). When new equalities are deduced, this option causes them to be used as rewrite rules.

Table 1: Results for Basic Theorems

	OTTER	Roo-8	Roo-12
Theorem 1: $x^2 = e$ Group			
proof time	0.20	0.32	0.32
generated	222	2300	1867
kept	13	30	40
memory (K)	31	728	564
Theorem 2: Commutator			
proof time	35.60	26.89	25.97
generated	20575	88838	131429
kept	4505	3684	1697
memory (K)	1564	12515	12670
Theorem 3: $x^2 = x$ Ring			
proof time	145.41	35.57	38.18
generated	56025	134744	221890
kept	13990	4316	2736
memory (K)	4342	14333	18739
Theorem 4: XGK			
proof time	407.50	159.87	55.37
generated	177109	663722	263233
kept	15320	16519	9466
memory (K)	8047	19539	22189
Theorem 5: Imp-4 (CD-67)			
proof time	7711.98	1051.55	909.95
generated	8341570	7171447	8182376
kept	17862	14855	17666
memory (K)	10729	13983	15098
Theorem 6: MV-1 (CD-57)			
proof time	17.68	4.37	14.71
generated	16687	24159	114051
kept	4837	1024	2000
memory (K)	2171	6479	12161
Theorem 7: MV-2 (CD-60)			
proof time	2184.96	427.89	152.53
generated	3214280	4311090	1997084
kept	16250	12374	10750
memory (K)	7216	13664	13755

Table 2: Results for Equality Problems

	OTTER	Roo-8	Roo-12
Theorem EQ-1: Commutator			
proof time	1.49	0.76	0.86
generated	542	1727	2144
kept	114	91	89
memory (K)	255	1208	1460
Theorem EQ-2: Robbins, $c + c = c$			
proof time	98.19	18.63	13.43
generated	50001	56067	59151
kept	4548	2450	1235
memory (K)	5652	12676	13342
Theorem EQ-3: TBA			
proof time	16.78	4.10	3.16
generated	3945	9307	11170
kept	1030	620	378
memory (K)	1564	4880	5043
Theorem EQ-4: Group single axiom			
proof time	44.12	10.56	9.25
generated	3417	11778	16118
kept	2507	1015	863
memory (K)	4470	13889	17110
Theorem EQ-5: Wajsberg algebra			
proof time	2248.86	425.99	491.67
generated	1012625	971543	1437272
kept	5897	4374	4022
memory (K)	6801	13376	14525

set(dynamic_demod_all). This option has OTTER use all new orientable equalities as rewrite rules.

set(index_for_back_demod). This option causes indexing to be used when searching for terms to which a new rewrite rule can be applied. Roo requires this “option” whenever back demodulation is enabled. OTTER frequently benefits from this option.

assign(pick_given_ratio,5). By default OTTER chooses each new given clause based on its symbol count. Hence, a heavy clause that is needed for the proof cannot be used until all lighter clauses have been used. Recently we have found it useful to mix this strategy with a breadth-first strategy by choosing some percentage of the given clauses according to the order in which they are generated rather than by weight. This setting chooses every sixth given clause in order of generation, and the rest by symbol count.

clear(print_kept). clear(print_new_demod).clear(print_back_demod). These options suppress output, saving file space and a little time.

assign(max_mem,20000). This setting restricts memory usage to 20 megabytes. Its real use is in conjunction with the next parameter.

set(control_memory). This setting has a relatively complex effect. Every ten given clauses, memory usage is analyzed. If more than a third of `max_mem` has been used, then the `max_weight` parameter is automatically set to a value calculated such that only the lightest 5% of the clauses in the current set of support have lower weight. No clauses are deleted, but from this point on, new clauses heavier than this weight are discarded. Using this parameter has the effect of allowing the system to choose a value for `max_weight` and adjust it during the run.

set(knuth_bendix). This option causes OTTER and Roo to automatically set a collection of options that approximate a Knuth-Bendix completion procedure. Under this option, the theorem prover orders equalities, paramodulates from left sides into left sides, and back demodulates.

set(process_input). This option causes all input clauses to be processed (subsumption, demodulation, equality ordering, back demodulation) as if they were generated clauses.

lex(*list of symbols*). This command specifies an ordering on constant, function, and predicate symbols, with smallest first. For the experiments described in this paper, the ordering is used to attempt to orient equalities.

set(lex_rpo). This option specifies the lexicographic recursive path ordering for comparing terms when attempting to orient equalities.

lrpo_lr_status(*list of symbols*). This command specifies that function symbols are to be compared left-to-right when applying the lexicographic recursive path ordering.

4 Failures on Equality Theorems 6–10

Theorem EQ-6. *The fragment $\{B, W, M\}$ of combinatory logic contains fixed point combinator.* OTTER found a proof, but the setting were different from those used in theorems EQ-1 through EQ-5. The important difference is that the initial set of

support consists of the denial only (so that all generated clauses are negative), and paramodulation is allowed into both arguments of equality literals. The following input file causes OTTER to find a proof of EQ-6 in about 27 seconds.

```

set(para_into).
clear(para_from_right).
set(order_eq).
assign(max_mem, 16000).
set(lex_rpo).
clear(print_kept).

lex([B,W,L,M,a(x,x),f(x)]).
lrpo_lr_status([a(x,x)]).

list(usables).
(x = x).
(a(a(a(B,x),y),z) = a(x,a(y,z))).
(a(a(W,x),y) = a(a(x,y),y)).
(a(M,x) = a(x,x)).
end_of_list.

list(sos).
(a(y,f(y)) != a(f(y),a(y,f(y)))) | $Ans(y).
end_of_list.

list(demodulators).
(a(a(a(B,x),y),z) = a(x,a(y,z))).
end_of_list.

```

Theorem EQ-7. *Rings in which $x^3 = x$ are commutative.* As far as we know, OTTER has never found a proof of this theorem, except with highly specialized settings and weight templates. We suspect that with associative-commutative unification, OTTER would be able to prove it.

Theorem EQ-8. *The fragment $\{B, W\}$ of combinatory logic contains fixed point combinator.* This theorem is much more difficult than EQ-6, and the strategy above that works for EQ-6 fails for EQ-8. The kernel method [7], which was developed for this type of problem, finds a proof of EQ-8 within a few seconds.

Theorems EQ-9 and EQ-10. On Moufang identities in nonassociative rings (EQ-9), and on right alternative nonassociative rings (EQ-10). The complicated definitions in these theorems cause terms in the conclusion to be greatly expanded. OTTER cannot cope with the complex conclusions, because it likes to focus on simple terms. As with (EQ-7), we believe that associative-commutative unification would be helpful for these theorems.

5 Summary of OTTER Outputs for the Basic Set

5.1 Theorem 1: $x^2 = e$ Groups are Commutative (P-form)

----- OTTER 2.2, July 1991 -----

The job began on altair.mcs.anl.gov, Wed Jun 3 13:15:19 1992
The command was "otter22".

```

set(hyper_res).
set(back_demod).
set(dynamic_demod_all).
assign(pick_given_ratio,5).
clear(print_kept).
assign(max_mem,20000).
set(control_memory).

list(usable).
1 [] -P(x,y,u) | -P(y,z,v) | -P(u,z,w) | P(x,v,w).
2 [] -P(x,y,u) | -P(y,z,v) | -P(x,v,w) | P(u,z,w).
3 [] -P(x,y,u) | -P(x,y,v) | eq(u,v).
4 [] eq(x,x).
5 [] -eq(x,y) | eq(y,x).
6 [] -eq(x,y) | -eq(y,z) | eq(x,z).
7 [] -eq(u,v) | -P(u,x,y) | P(v,x,y).
8 [] -eq(u,v) | -P(x,u,y) | P(x,v,y).
9 [] -eq(u,v) | -P(x,y,u) | P(x,y,v).
10 [] -eq(u,v) | eq(f(u,x),f(v,x)).
11 [] -eq(u,v) | eq(f(x,u),f(x,v)).
12 [] -eq(u,v) | eq(g(u),g(v)).
end_of_list.

list(sos).
13 [] P(e,x,x).
14 [] P(x,e,x).
15 [] P(g(x),x,e).
16 [] P(x,g(x),e).
17 [] P(x,y,f(x,y)).
18 [] P(x,x,e).
19 [] P(a,b,c).
20 [] -P(b,a,c).
end_of_list.

OTTER sets dynamic_demod, because back_demod is set.
OTTER sets order_eq, because dynamic_demod is set.

----> UNIT CONFLICT at 0.20 sec ----> 35 [binary,34,20] .
Level of proof is 3, length is 4.

----- PROOF -----
1 [] -P(x,y,u) | -P(y,z,v) | -P(u,z,w) | P(x,v,w).
2 [] -P(x,y,u) | -P(y,z,v) | -P(x,v,w) | P(u,z,w).
13 [] P(e,x,x).
14 [] P(x,e,x).
18 [] P(x,x,e).
19 [] P(a,b,c).
20 [] -P(b,a,c).
21 [hyper,19,2,18,14] P(c,b,a).
22 [hyper,19,1,18,13] P(a,c,b).
23 [hyper,21,1,18,13] P(c,a,b).

```

```

34 [hyper,23,2,22,19] P(b,a,c).
35 [binary,34,20] .

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

----- statistics -----  

clauses input          20  

clauses given           9  

clauses generated      222  

demod & eval rewrites   26  

tautologies deleted     0  

clauses forward subsumed 209  

  (subsumed by sos)    13  

clauses kept            13  

new demodulators        1  

empty clauses            1  

clauses back demodulated 0  

clauses back subsumed    0  

sos size                 12  

Kbytes malloced         31  

----- times (seconds) -----  

run time                0.22          (run time 0 hr, 0 min, 0 sec)  

system time              0.11  

input time               0.02  

  classify time          0.00  

hyper_res time           0.05  

pre_process time         0.09  

  demod time             0.01  

  weigh cl time          0.00  

  for_sub time            0.05  

  renumber time           0.00  

  keep cl time            0.00  

  print_cl time           0.00  

  conflict time           0.00  

post_process time        0.01  

  back demod time         0.00  

  back_sub time            0.01  

lex_rpo time              0.00  

The job finished       Wed Jun  3 13:15:19 1992

```

5.2 Theorem 2: The Commutator Theorem (P-form)

----- OTTER 2.2, July 1991 -----
The job began on altair.mcs.anl.gov, Wed Jun 3 13:13:02 1992
The command was "otter22".

```

set(hyper_res).
set(back_demod).
set(dynamic_demod_all).
assign(pick_given_ratio,5).
clear(print_kept).
assign(max_mem,20000).

```

```

set(control_memory).

list(usable).
1 [] -P(x,y,u) | -P(y,z,v) | -P(u,z,w) | P(x,v,w).
2 [] -P(x,y,u) | -P(y,z,v) | -P(x,v,w) | P(u,z,w).
3 [] -P(x,y,u) | -P(x,y,v) | eq(u,v).
4 [] eq(x,x).
5 [] -eq(x,y) | eq(y,x).
6 [] -eq(x,y) | -eq(y,z) | eq(x,z).
7 [] -eq(u,v) | -P(u,x,y) | P(v,x,y).
8 [] -eq(u,v) | -P(x,u,y) | P(x,v,y).
9 [] -eq(u,v) | -P(x,y,u) | P(x,y,v).
10 [] -eq(u,v) | eq(f(u,x),f(v,x)).
11 [] -eq(u,v) | eq(f(x,u),f(x,v)).
12 [] -eq(u,v) | eq(g(u),g(v)).
13 [] -P(x,x,y) | P(x,y,e).
14 [] -P(x,x,y) | P(y,x,e).
end_of_list.

list(sos).
15 [] P(e,x,x).
16 [] P(x,e,x).
17 [] P(g(x),x,e).
18 [] P(x,g(x),e).
19 [] P(x,y,f(x,y)).
20 [] P(a,b,c).
21 [] P(c,g(a),d).
22 [] P(d,g(b),h).
23 [] P(h,b,j).
24 [] P(j,g(h),k).
25 [] -P(k,g(b),e).
end_of_list.

OTTER sets dynamic_demod, because back_demod is set.
OTTER sets order_eq, because dynamic_demod is set.

-----> UNIT CONFLICT at 35.60 sec -----> 4648 [binary,4647,49] .
Level of proof is 15, length is 37.

----- PROOF -----
1 [] -P(x,y,u) | -P(y,z,v) | -P(u,z,w) | P(x,v,w).
2 [] -P(x,y,u) | -P(y,z,v) | -P(x,v,w) | P(u,z,w).
3 [] -P(x,y,u) | -P(x,y,v) | eq(u,v).
9 [] -eq(u,v) | -P(x,y,u) | P(x,y,v).
13 [] -P(x,x,y) | P(x,y,e).
14 [] -P(x,x,y) | P(y,x,e).
15 [] P(e,x,x).
16 [] P(x,e,x).
17 [] P(g(x),x,e).
18 [] P(x,g(x),e).
19 [] P(x,y,f(x,y)).
20 [] P(a,b,c).
21 [] P(c,g(a),d).
22 [] P(d,g(b),h).

```

```

23 [] P(h,b,j).
24 [] P(j,g(h),k).
25 [] -P(k,g(b),e).
28 [hyper,17,2,17,16] P(e,x,g(g(x))).
37 [hyper,21,2,17,16] P(d,a,c).
39 [hyper,37,1,17,15] P(g(d),c,a).
41 [hyper,22,2,17,16] P(h,b,d).
45,44 [hyper,41,3,23] eq(j,d).
46 [hyper,41,1,17,15] P(g(h),d,b).
47 [back_demod,24,demod,45] P(d,g(h),k).
48 [hyper,19,14] P(f(x,x),x,e).
49 [hyper,19,13] P(x,f(x,x),e).
65,64 [hyper,19,3,16] eq(f(x,e),x).
67,66 [hyper,19,3,15] eq(f(e,x),x).
79 [hyper,19,2,19,19] P(f(x,y),z,f(x,f(y,z))).
80 [hyper,19,2,19,18,demod,65] P(f(x,y),g(y),x).
160 [hyper,39,2,18,19,demod,65] P(a,g(c),g(d)).
176 [hyper,46,2,18,19,demod,65] P(b,g(d),g(h)).
183,182 [hyper,28,3,19,demod,67] eq(g(g(x)),x).
192 [hyper,47,2,18,19,demod,183,65] P(k,h,d).
312 [hyper,48,1,20,19,demod,67] P(f(a,a),c,b).
317,316 [hyper,49,3,19] eq(f(x,f(x,x)),e).
319 [hyper,49,2,192,19,demod,65] P(d,f(h,h),k).
497 [hyper,176,3,19] eq(f(b,g(d)),g(h)).
705 [hyper,312,1,19,19] P(a,f(a,c),b).
715,714 [hyper,319,3,19] eq(f(d,f(h,h)),k).
932 [hyper,705,2,37,19] P(c,f(a,c),f(d,b)).
1180 [hyper,80,1,49,19,demod,67] P(x,x,g(x)).
1235,1234 [hyper,1180,3,19] eq(g(x),f(x,x)).
1256 [hyper,1180,2,19,160,demod,1235] P(f(a,c),c,f(d,d)).
1481,1480 [back_demod,497,demod,1235,1235] eq(f(b,f(d,d)),f(h,h)).
1534 [back_demod,25,demod,1235] -P(k,f(b,b),e).
1586,1585 [hyper,79,3,19] eq(f(f(x,y),z),f(x,f(y,z))).
3340,3339 [hyper,932,3,19] eq(f(c,f(a,c)),f(d,b)).
3736 [hyper,1256,2,48,19,demod,1586,3340,1586,1586,1481,715] P(e,c,f(a,k)).
3775 [hyper,3736,3,19,demod,67] eq(f(a,k),c).
3794 [hyper,3775,9,19] P(a,k,c).
3817 [hyper,3794,2,79,312,demod,317] P(e,k,b).
3959,3958 [hyper,3817,3,19,demod,67] eq(k,b).
4647 [back_demod,1534,demod,3959] -P(b,f(b,b),e).
4648 [binary,4647,49] .

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

----- statistics -----

clauses input	25
clauses given	165
clauses generated	20575
demod & eval rewrites	32696
tautologies deleted	0
clauses forward subsumed	18332
(subsumed by sos)	6484
clauses kept	4505
new demodulators	117

```

empty clauses          1
clauses back demodulated   2262
clauses back subsumed      60
sos size                  2089
Kbytes malloced           1564

----- times (seconds) -----
run time            35.65          (run time 0 hr, 0 min, 35 sec)
system time         6.88
input time          0.01
  classify time     0.00
hyper_res time      6.07
pre_process time    19.21
  demod time        6.33
  weigh cl time    0.00
  for_sub time      6.37
  renumber time     0.94
  keep cl time      2.18
  print_cl time     0.00
  conflict time     1.48
post_process time   9.71
  back demod time   7.73
  back_sub time     1.82
lex_rpo time        0.00
The job finished      Wed Jun 3 13:13:44 1992

```

5.3 Theorem 3: $x^2 = x$ Rings are Commutative (P-form)

```

----- OTTER 2.2, July 1991 -----
The job began on altair.mcs.anl.gov, Wed Jun 3 13:19:36 1992
The command was "otter22".

```

```

set(hyper_res).
set(back_demod).
set(dynamic_demod_all).
assign(pick_given_ratio,5).
clear(print_kept).
assign(max_mem,20000).
set(control_memory).

list(usable).
1 [] -S(x,y,u) | -S(y,z,v) | -S(u,z,w) | S(x,v,w).
2 [] -S(x,y,u) | -S(y,z,v) | -S(x,v,w) | S(u,z,w).
3 [] -S(x,y,u) | -S(x,y,v) | eq(u,v).
4 [] eq(x,x).
5 [] -eq(x,y) | eq(y,x).
6 [] -eq(x,y) | -eq(y,z) | eq(x,z).
7 [] -eq(u,v) | -S(u,x,y) | S(v,x,y).
8 [] -eq(u,v) | -S(x,u,y) | S(x,v,y).
9 [] -eq(u,v) | -S(x,y,u) | S(x,y,v).
10 [] -eq(u,v) | eq(j(u,x),j(v,x)).
11 [] -eq(u,v) | eq(j(x,u),j(x,v)).
12 [] -eq(u,v) | eq(g(u),g(v)).

```

```

13 [] -S(x,y,z) | S(y,x,z).
14 [] -P(x,y,u) | -P(y,z,v) | -P(u,z,w) | P(x,v,w).
15 [] -P(x,y,u) | -P(y,z,v) | -P(x,v,w) | P(u,z,w).
16 [] -P(x,y,v1) | -P(x,z,v2) | -S(y,z,v3) | -P(x,v3,v4) | S(v1,v2,v4).
17 [] -P(x,y,v1) | -P(x,z,v2) | -S(y,z,v3) | -S(v1,v2,v4) | P(x,v3,v4).
18 [] -P(y,x,v1) | -P(z,x,v2) | -S(y,z,v3) | -P(v3,x,v4) | S(v1,v2,v4).
19 [] -P(y,x,v1) | -P(z,x,v2) | -S(y,z,v3) | -S(v1,v2,v4) | P(v3,x,v4).
20 [] -P(x,y,u) | -P(x,y,v) | eq(u,v).
21 [] -eq(u,v) | -P(u,x,y) | P(v,x,y).
22 [] -eq(u,v) | -P(x,u,y) | P(x,v,y).
23 [] -eq(u,v) | -P(x,y,u) | P(x,y,v).
24 [] -eq(u,v) | eq(f(u,x),f(v,x)).
25 [] -eq(u,v) | eq(f(x,u),f(x,v)).
end_of_list.

list(sos).
26 [] S(0,x,x).
27 [] S(x,0,x).
28 [] S(g(x),x,0).
29 [] S(x,g(x),0).
30 [] S(x,y,j(x,y)).
31 [] P(0,x,0).
32 [] P(x,0,0).
33 [] P(x,y,f(x,y)).
34 [] P(x,x,x).
35 [] P(a,b,c).
36 [] -P(b,a,c).
end_of_list.

OTTER sets dynamic_demod, because back_demod is set.
OTTER sets order_eq, because dynamic_demod is set.

----> UNIT CONFLICT at 145.41 sec ----> 14124 [binary,14123,36] .
Level of proof is 16, length is 41.

```

----- PROOF -----

```

1 [] -S(x,y,u) | -S(y,z,v) | -S(u,z,w) | S(x,v,w).
2 [] -S(x,y,u) | -S(y,z,v) | -S(x,v,w) | S(u,z,w).
3 [] -S(x,y,u) | -S(x,y,v) | eq(u,v).
13 [] -S(x,y,z) | S(y,x,z).
14 [] -P(x,y,u) | -P(y,z,v) | -P(u,z,w) | P(x,v,w).
15 [] -P(x,y,u) | -P(y,z,v) | -P(x,v,w) | P(u,z,w).
16 [] -P(x,y,v1) | -P(x,z,v2) | -S(y,z,v3) | -P(x,v3,v4) | S(v1,v2,v4).
17 [] -P(x,y,v1) | -P(x,z,v2) | -S(y,z,v3) | -S(v1,v2,v4) | P(x,v3,v4).
18 [] -P(y,x,v1) | -P(z,x,v2) | -S(y,z,v3) | -P(v3,x,v4) | S(v1,v2,v4).
19 [] -P(y,x,v1) | -P(z,x,v2) | -S(y,z,v3) | -S(v1,v2,v4) | P(v3,x,v4).
20 [] -P(x,y,u) | -P(x,y,v) | eq(u,v).
23 [] -eq(u,v) | -P(x,y,u) | P(x,y,v).
26 [] S(0,x,x).
27 [] S(x,0,x).
28 [] S(g(x),x,0).
29 [] S(x,g(x),0).
30 [] S(x,y,j(x,y)).
31 [] P(0,x,0).

```

```

32 [] P(x,0,0).
33 [] P(x,y,f(x,y)).
34 [] P(x,x,x).
35 [] P(a,b,c).
36 [] -P(b,a,c).
37 [hyper,35,15,35,34] P(c,b,c).
38 [hyper,35,14,34,35] P(a,c,c).
43,42 [hyper,28,3,27] eq(g(0),0).
44 [hyper,28,2,28,27] S(0,x,g(g(x))).
63 [hyper,30,19,34,37,30] P(j(b,c),b,j(b,c)).
92 [hyper,30,17,38,34,30] P(a,j(c,a),j(c,a)).
120 [hyper,30,13] S(x,y,j(y,x)).
126,125 [hyper,30,3,27] eq(j(x,0),x).
128,127 [hyper,30,3,26] eq(j(0,x),x).
131 [hyper,30,2,30,28,demod,126] S(j(x,g(y)),y,x).
139 [hyper,30,1,28,30,demod,128] S(g(x),j(x,y),y).
150,149 [hyper,33,20,32] eq(f(x,0),0).
152,151 [hyper,33,20,31] eq(f(0,x),0).
193 [hyper,33,18,34,33,29,demod,152] S(x,f(g(x),x),0).
248 [hyper,33,16,34,33,29,demod,150] S(x,f(x,g(x)),0).
258 [hyper,33,16,33,34,28,demod,150] S(f(x,g(x)),x,0).
325,324 [hyper,44,3,30,demod,128] eq(g(g(x)),x).
450 [hyper,120,2,120,28,demod,128] S(j(g(x),y),x,y).
475 [hyper,120,1,28,120,demod,126] S(g(x),j(y,x),y).
1477 [hyper,139,3,30] eq(j(g(x),j(x,y)),y).
1905 [hyper,193,2,29,131,demod,43,126,325,43,126] S(0,f(x,g(x)),x).
2976 [hyper,248,3,30] eq(j(x,f(x,g(x))),0).
3558 [hyper,258,2,30,131,demod,43,126] S(j(x,f(y,g(y))),y,x).
7630,7629 [hyper,1905,3,120,demod,126] eq(f(x,g(x)),x).
7631 [hyper,1905,2,450,248,demod,43,128,43,128,7630,43,128,43,128,7630,7630] S(x,x,0).
7632 [hyper,1905,1,475,248,demod,128] S(g(x),0,x).
7636 [hyper,1905,1,258,120,demod,7630,7630] S(x,j(y,x),y).
7637 [hyper,1905,1,258,30,demod,7630,7630] S(x,j(x,y),y).
7793 [back_demod,3558,demod,7630] S(j(x,y),y,x).
7836,7835 [back_demod,2976,demod,7630] eq(j(x,x),0).
8850 [hyper,63,17,34,120,120,demod,7836] P(j(b,c),j(b,j(b,c)),0).
9362,9361 [hyper,7632,3,120,demod,128] eq(g(x),x).
10116,10115 [back_demod,1477,demod,9362] eq(j(x,j(x,y)),y).
10286 [back_demod,8850,demod,10116] P(j(b,c),c,0).
10433 [hyper,92,19,34,7793,7631] P(c,j(c,a),0).
11007 [hyper,10286,19,34,7636,120,demod,128] P(b,c,c).
11206 [hyper,11007,15,33,35] P(f(b,a),b,c).
11469 [hyper,10433,17,34,7637,120,demod,128] P(c,a,c).
12377 [hyper,11206,15,33,34] P(c,a,f(b,a)).
14119 [hyper,12377,20,11469] eq(f(b,a),c).
14123 [hyper,14119,23,33] P(b,a,c).
14124 [binary,14123,36] .

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

```

----- statistics -----
clauses input          36
clauses given          120
clauses generated      56025
```

```

demod & eval rewrites      77893
tautologies deleted          1
clauses forward subsumed    53529
  (subsumed by sos)        19775
clauses kept                13990
new demodulators            97
empty clauses                 1
clauses back demodulated   11495
clauses back subsumed       39
sos size                     2410
Kbytes malloced             4342

----- times (seconds) -----
run time           145.46          (run time 0 hr, 2 min, 25 sec)
system time        20.87
input time         0.04
  claусify time     0.00
hyper_res time    20.44
pre_process time  52.21
  demod time        12.31
  weigh cl time    0.00
  for_sub time     18.56
  renumber time    2.96
  keep cl time     7.94
  print_cl time    0.00
  conflict time    5.10
post_process time 71.27
  back demod time  63.58
  back_sub time    7.18
lex_rpo time       0.00
The job finished      Wed Jun 3 13:22:23 1992

```

5.4 Theorem 4: Equivalential Calculus, XGK → PYO

```

----- OTTER 2.2, July 1991 -----
The job began on altair.mcs.anl.gov, Wed Jun 3 13:22:37 1992
The command was "otter22".

```

```

set(hyper_res).
set(back_demod).
set(dynamic_demod_all).
assign(pick_given_ratio,5).
clear(print_kept).
assign(max_mem,20000).
set(control_memory).

list(usable).
1 [] -P(x) | -P(e(x,y)) | P(y).
end_of_list.

list(sos).
2 [] P(e(x,e(e(y,e(z,x)),e(z,y)))). 
3 [] -P(e(e(e(a,e(b,c)),c),e(b,a))). 

```

```

end_of_list.
OTTER sets dynamic_demod, because back_demod is set.
OTTER sets order_eq, because dynamic_demod is set.

Resetting weight limit to 20.
----> UNIT CONFLICT at 407.50 sec ----> 15324 [binary,15323,3] .
Level of proof is 13, length is 19.

```

----- PROOF -----

```

1 [] -P(x) | -P(e(x,y)) | P(y).
2 [] P(e(x,e(e(y,e(z,x)),e(z,y)))). 
3 [] -P(e(e(e(a,e(b,c)),c),e(b,a))). 
4 [hyper,2,1,2] P(e(e(x,e(y,e(z,e(e(u,e(v,z)),e(v,u))))),e(y,x))). 
6 [hyper,4,1,2] P(e(e(e(x,e(y,z)),e(y,x)),e(z,u)),u)). 
8 [hyper,6,1,6] P(e(x,x)). 
9 [hyper,6,1,4] P(e(e(x,e(e(y,e(z,x)),e(z,y))),e(u,u))). 
13 [hyper,8,1,2] P(e(e(x,e(y,e(z,z))),e(y,x))). 
18 [hyper,13,1,2] P(e(e(x,e(x,y)),y)). 
21 [hyper,13,1,2] P(e(e(x,e(y,e(e(z,e(u,e(v,v))),e(u,z)))),e(y,x))). 
39 [hyper,18,1,13] P(e(x,e(y,e(x,e(z,z)))))). 
42 [hyper,18,1,2] P(e(e(x,e(y,e(e(z,e(z,u)),u))),e(y,x))). 
108 [hyper,39,1,4] P(e(x,e(y,e(y,x)))). 
133 [hyper,108,1,2] P(e(e(x,e(y,e(z,e(u,e(u,z))))),e(y,x))). 
146 [hyper,9,1,2] P(e(e(x,e(y,e(e(z,e(e(u,e(v,z)),e(v,u))),e(w,w)))),e(y,x))). 
682 [hyper,42,1,18] P(e(x,e(y,e(x,e(e(z,e(z,u)),u)))))). 
2253 [hyper,133,1,2] P(e(e(e(x,e(x,y)),e(y,z)),z)). 
8738 [hyper,682,1,4] P(e(x,e(y,e(z,e(z,y)),x)))). 
8897 [hyper,8738,1,2253] P(e(e(x,e(x,y)),e(z,e(z,y)))). 
9048 [hyper,8897,1,21] P(e(e(x,e(y,e(z,z))),e(u,e(u,e(y,x))))). 
13855 [hyper,9048,1,4] P(e(x,e(e(y,z),e(e(z,e(y,x)),e(u,u))))). 
15323 [hyper,13855,1,146] P(e(e(e(x,e(y,z)),z),e(y,x))). 
15324 [binary,15323,3] .

```

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

----- statistics -----

clauses input	3
clauses given	587
clauses generated	177109
demod & eval rewrites	0
clauses wt,lit,sk delete	102987
tautologies deleted	0
clauses forward subsumed	58802
(subsumed by sos)	12239
clauses kept	15320
new demodulators	0
empty clauses	1
clauses back demodulated	0
clauses back subsumed	0
sos size	14735
Kbytes malloced	8047

----- times (seconds) -----

```

run time          407.53           (run time 0 hr, 6 min, 47 sec)
system time       41.90
input time        0.01
  classify time   0.00
hyper_res time    72.62
pre_process time 100.67
  demod time     0.00
  weigh cl time  15.72
  for_sub time   18.83
  renumber time   18.15
  keep cl time   21.08
  print_cl time  0.00
  conflict time  4.38
post_process time 223.50
  back demod time 0.00
  back_sub time  223.01
lex_rpo time      0.00
The job finished      Wed Jun  3 13:30:09 1992

```

5.5 Theorem 5: Implicational Calculus Single Axiom, CD-67 (Imp-4)

----- OTTER 2.2, July 1991 -----
 The job began on altair.mcs.anl.gov, Wed Jun 3 14:27:00 1992
 The command was "otter22".

```

set(hyper_res).
set(back_demod).
set(dynamic_demod_all).
clear(print_kept).
assign(pick_given_ratio,5).
assign(max_mem,20000).
set(control_memory).

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

list(sos).
2 [] P(i(i(i(x,y),z),i(i(z,x),i(u,x)))). 
3 [] -P(i(i(a,b),i(i(b,c),i(a,c)))). 
end_of_list.

OTTER sets dynamic_demod, because back_demod is set.
OTTER sets order_eq, because dynamic_demod is set.

Resetting weight limit to 20.
Resetting weight limit to 18.
Resetting weight limit to 16.
----> UNIT CONFLICT at 7711.98 sec ----> 17866 [binary,17865,3] .
Level of proof is 40, length is 94.

----- PROOF -----
1 [] -P(x) | -P(i(x,y)) | P(y).

```

```

2 [] P(i(i(i(x,y),z),i(z,x),i(u,x))).
3 [] -P(i(i(a,b),i(i(b,c),i(a,c))).
4 [hyper,2,1,2] P(i(i(i(x,y),i(z,y)),i(y,u)),i(v,i(y,u))).
5 [hyper,4,1,4] P(i(x,i(i(y,z),i(z,i(y,z))))).
6 [hyper,4,1,2] P(i(i(i(x,i(y,z)),i(i(u,y),i(v,y))),i(w,i(i(u,y),i(v,y))))).
7 [hyper,5,1,5] P(i(i(x,y),i(y,i(x,y))).
12 [hyper,7,1,2] P(i(i(i(x,i(y,x)),y),i(z,y))).
15 [hyper,6,1,12] P(i(x,i(i(y,z),i(z,z))).
19 [hyper,6,1,2] P(i(i(i(x,i(i(y,z),i(u,z))),i(v,i(z,w))),i(v6,i(v,i(z,w))))).
20 [hyper,15,1,15] P(i(i(x,y),i(y,y))).
23 [hyper,20,1,12] P(i(i(x,y),i(x,y))).
24 [hyper,20,1,2] P(i(i(i(x,x),y),i(z,y))).
26 [hyper,23,1,2] P(i(i(i(x,y),x),i(z,x))).
27 [hyper,24,1,24] P(i(x,i(y,i(z,z))).
28 [hyper,24,1,23] P(i(x,i(y,y))).
31 [hyper,24,1,4] P(i(x,i(y,i(z,y))).
32 [hyper,28,1,28] P(i(x,x)).
38 [hyper,31,1,32] P(i(x,i(y,x))).
41 [hyper,38,1,27] P(i(x,i(y,i(z,i(u,u))))).
45 [hyper,38,1,2] P(i(x,i(i(i(y,z),u),i(i(u,y),i(v,y))))).
47 [hyper,38,1,2] P(i(i(i(x,i(y,z)),y),i(u,y))).
48 [hyper,26,1,2] P(i(i(i(x,y),i(y,z)),i(u,i(y,z))))).
53 [hyper,41,1,38] P(i(x,i(y,i(z,i(u,i(v,v)))))).
61 [hyper,19,1,2] P(i(x,i(i(i(i(y,z),i(u,z)),v),i(z,v))))..
80 [hyper,47,1,2] P(i(i(i(x,y),i(z,i(y,u))),i(v,i(z,i(y,u))))).
85 [hyper,53,1,38] P(i(x,i(y,i(z,i(u,i(v,i(w,w))))))).
92 [hyper,48,1,2] P(i(i(i(x,i(y,z)),i(u,y)),i(v,i(u,y))).
122 [hyper,45,1,2] P(i(i(i(i(i(x,y),z),i(i(z,x),i(u,x))),v),i(w,v))).
130 [hyper,61,1,85] P(i(i(i(i(x,y),i(z,y)),u),i(y,u))).
138 [hyper,130,1,26] P(i(x,i(y,i(z,x))).
139 [hyper,130,1,2] P(i(x,i(i(i(y,x),z),i(u,z))).
166 [hyper,138,1,2] P(i(i(i(x,i(y,i(z,u))),z),i(v,z))).
338 [hyper,92,1,2] P(i(x,i(i(i(y,z),u),z),i(y,z))).
353 [hyper,338,1,338] P(i(i(i(i(x,y),z),y),i(x,y))).
362 [hyper,353,1,2] P(i(i(i(x,y),i(i(x,y),z)),i(u,i(i(x,y),z))).
970 [hyper,362,1,47] P(i(x,i(i(i(y,i(z,u)),z),z))).
973 [hyper,362,1,26] P(i(x,i(i(i(y,z),y),y))).
974 [hyper,362,1,24] P(i(x,i(i(i(y,y),z),z))).
991 [hyper,973,1,973] P(i(i(i(x,y),x),x)).
1004 [hyper,991,1,2] P(i(i(x,i(x,y)),i(z,i(x,y))).
1005 [hyper,974,1,991] P(i(i(i(x,x),y),y)).
1016 [hyper,970,1,1005] P(i(i(i(x,i(y,z)),y),y)).
1027 [hyper,1016,1,2] P(i(i(x,i(y,i(x,z))),i(u,i(y,i(x,z))))).
1072 [hyper,1004,1,1004] P(i(x,i(i(y,i(y,z)),i(y,z))).
1083 [hyper,1072,1,1072] P(i(i(x,i(x,y)),i(x,y))).
1104 [hyper,1083,1,166] P(i(i(i(x,i(y,i(z,u))),z),z)).
1114 [hyper,1083,1,92] P(i(i(i(x,i(y,z)),i(u,y)),i(u,y))).
1118 [hyper,1083,1,80] P(i(i(i(x,y),i(z,i(y,u))),i(z,i(y,u))).
1124 [hyper,1083,1,48] P(i(i(i(x,y),i(y,z)),i(y,z))).
1155 [hyper,1104,1,2] P(i(i(x,i(y,i(z,i(x,u)))),i(v,i(y,i(z,i(x,u)))))).
1177 [hyper,1124,1,2] P(i(i(i(x,y),i(z,x)),i(u,i(z,x))).
1374 [hyper,1177,1,1083] P(i(i(i(x,y),i(z,x)),i(z,x))).
1565 [hyper,1027,1,139] P(i(x,i(i(i(y,z),u),i(z,u))).
1566 [hyper,1027,1,2] P(i(x,i(i(y,z),i(i(i(z,u),y),z))).

```

```

1567 [hyper,1027,1,1083] P(i(i(x,i(y,i(x,z))),i(y,i(x,z)))).  

1577 [hyper,1565,1,1565] P(i(i(i(x,y),z),i(y,z))).  

1588 [hyper,1577,1,2] P(i(x,i(i(x,y),i(z,y)))).  

1592 [hyper,1577,1,122] P(i(x,i(y,i(i(y,z),i(u,z))))).  

1645 [hyper,1588,1,2] P(i(i(i(i(x,y),z),i(u,z)),x),i(v,x))).  

1661 [hyper,1566,1,1592] P(i(i(x,y),i(i(i(y,z),x),y))).  

1672 [hyper,1661,1,1588] P(i(i(i(i(i(x,y),i(z,y)),u),x),i(i(x,y),i(z,y)))).  

1703 [hyper,1661,1,1004] P(i(i(i(i(x,i(y,z)),u),i(y,i(y,z))),i(x,i(y,z)))).  

1741 [hyper,1661,1,138] P(i(i(i(i(x,i(y,z)),u),z),i(x,i(y,z)))).  

1762 [hyper,1661,1,47] P(i(i(i(i(x,y),z),i(i(u,i(y,v)),y)),i(x,y))).  

1765 [hyper,1661,1,26] P(i(i(i(i(x,y),z),i(i(y,u),y)),i(x,y))).  

2492 [hyper,1645,1,1083] P(i(i(i(i(i(x,y),z),i(u,z)),x),x)).  

4636 [hyper,1762,1,2492] P(i(i(i(i(i(x,i(y,z),u),i(y,z)),v),z),i(y,z))).  

7184 [hyper,1155,1,1083] P(i(i(x,i(y,i(z,i(x,u)))),i(y,i(z,i(x,u))))).  

10842 [hyper,4636,1,1765] P(i(i(i(x,i(i(i(y,z),y),u)),i(i(y,z),y)),y)).  

10924 [hyper,10842,1,1672] P(i(i(x,y),i(i(i(x,z),x),y))).  

10927 [hyper,10924,1,1588] P(i(i(i(x,y),x),i(i(x,z),i(u,z)))).  

10951 [hyper,10927,1,1741] P(i(x,i(y,i(i(x,z),i(u,z)))).  

10953 [hyper,10927,1,1703] P(i(x,i(i(x,y),y))).  

11237 [hyper,10953,1,10924] P(i(i(i(x,y),x),i(i(x,z),z))).  

11252 [hyper,10953,1,1661] P(i(i(i(i(x,y),y),z),x),i(i(x,y),y))).  

11310 [hyper,10951,1,1577] P(i(x,i(y,i(i(i(z,x),u),i(v,u)))).  

11344 [hyper,11237,1,1741] P(i(x,i(y,i(i(x,z),z)))).  

11355 [hyper,11237,1,2] P(i(i(i(i(x,y),y),i(x,z)),i(u,i(x,z)))).  

11414 [hyper,11344,1,1577] P(i(x,i(y,i(i(i(z,x),u),u)))).  

12034 [hyper,11355,1,1083] P(i(i(i(i(x,y),y),i(x,z)),i(x,z))).  

12131 [hyper,12034,1,11414] P(i(x,i(i(i(y,i(i(x,z),z)),u),u))).  

12134 [hyper,12034,1,11310] P(i(x,i(i(i(y,i(i(x,z),z)),u),i(v,u)))).  

12136 [hyper,12034,1,10951] P(i(x,i(i(i(i(x,y),y),z),i(u,z)))).  

12188 [hyper,12131,1,1374] P(i(i(i(x,i(i(i(y,z),u),u)),y),y)).  

12191 [hyper,12131,1,1114] P(i(i(i(x,i(i(i(y,i(z,u)),v),v)),z),z)).  

12238 [hyper,12136,1,1567] P(i(i(i(i(x,y),y),z),i(x,z))).  

12442 [hyper,12188,1,11252] P(i(i(x,i(i(i(x,y),z),z)),i(i(i(x,y),z),z))).  

13088 [hyper,12134,1,1118] P(i(i(i(x,i(i(i(y,z),u),u)),v),i(z,v))).  

13109 [hyper,12191,1,1672] P(i(i(x,y),i(i(i(z,i(x,u)),y),y))).  

13927 [hyper,13088,1,1672] P(i(i(i(x,y),z),i(i(i(u,x),z),z))).  

14592 [hyper,12442,1,13109] P(i(i(i(i(x,y),i(x,z)),y),y)).  

14632 [hyper,14592,1,13927] P(i(i(i(x,i(i(y,z),i(y,u))),z),z)).  

14829 [hyper,14632,1,1672] P(i(i(x,i(y,z)),i(i(y,x),i(y,z)))).  

15113 [hyper,14829,1,10951] P(i(i(x,y),i(x,i(i(y,z),i(u,z))))).  

16490 [hyper,15113,1,12238] P(i(x,i(i(x,y),i(i(y,z),i(u,z))))).  

17865 [hyper,16490,1,7184] P(i(i(x,y),i(i(y,z),i(x,z)))).  

17866 [binary,17865,3] .

```

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

----- statistics -----
clauses input 3
clauses given 3096
clauses generated 8341570
demod & eval rewrites 0
clauses wt,lit,sk delete 2972221
tautologies deleted 0
clauses forward subsumed 5351487

```

    (subsumed by sos)          83872
clauses kept                  17862
new demodulators                 0
empty clauses                     1
clauses back demodulated        0
clauses back subsumed           386
sos size                         14449
Kbytes malloced                  10729

----- times (seconds) -----
run time             7714.75          (run time 2 hr, 8 min, 34 sec)
system time          1863.38
input time            0.01
  classify time       0.00
hyper_res time        2863.86
pre_process time     4227.72
  demod time          0.00
  weigh cl time      736.90
  for_sub time        1461.46
  renumber time       866.73
  keep cl time        33.44
  print_cl time       0.00
  conflict time       4.74
post_process time    455.12
  back demod time    0.00
  back_sub time       450.01
lex_rpo time          0.00
The job finished      Wed Jun 3 17:07:15 1992

```

5.6 Theorem 6: Many-valued Sentential Calculus, CD-57

----- OTTER 2.2, July 1991 -----
The job began on altair.mcs.anl.gov, Wed Jun 3 13:16:52 1992
The command was "otter22".

```

set(hyper_res).
set(back_demod).
set(dynamic_demod_all).
clear(print_kept).
assign(pick_given_ratio,5).
assign(max_mem,20000).
set(control_memory).

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

list(sos).
2 [] P(i(x,i(y,x))).
3 [] P(i(i(x,y),i(i(y,z),i(x,z)))).
4 [] P(i(i(i(x,y),y),i(i(y,x),x))).
5 [] P(i(i(n(x),n(y)),i(y,x))).
6 [] -P(i(i(a,b),i(i(c,a),i(c,b)))).


```

```

end_of_list.
OTTER sets dynamic_demod, because back_demod is set.
OTTER sets order_eq, because dynamic_demod is set.

----> UNIT CONFLICT at 17.68 sec ----> 4844 [binary,4843,6] .
Level of proof is 4, length is 5.

```

----- PROOF -----

```

1 [] -P(x) | -P(i(x,y)) | P(y).
2 [] P(i(x,i(y,x))).
3 [] P(i(i(x,y),i(i(y,z),i(x,z)))).
4 [] P(i(i(i(x,y),y),i(i(y,x),x))).
6 [] -P(i(i(a,b),i(i(c,a),i(c,b)))). 
14 [hyper,3,1,3] P(i(i(i(i(x,y),i(z,y)),u),i(i(z,x),u))).
15 [hyper,3,1,2] P(i(i(i(x,y),z),i(y,z))).
24 [hyper,4,1,15] P(i(x,i(i(x,y),y))).
49 [hyper,24,1,3] P(i(i(i(i(x,y),y),z),i(x,z))).
4843 [hyper,49,1,14] P(i(i(x,y),i(i(z,x),i(z,y)))). 
4844 [binary,4843,6] .

```

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

----- statistics -----

clauses input	6
clauses given	181
clauses generated	16687
demod & eval rewrites	0
tautologies deleted	0
clauses forward subsumed	11850
(subsumed by sos)	1683
clauses kept	4837
new demodulators	0
empty clauses	1
clauses back demodulated	0
clauses back subsumed	11
sos size	4650
Kbytes malloced	2171

----- times (seconds) -----

run time	17.69	(run time 0 hr, 0 min, 17 sec)
system time	4.34	
input time	0.02	
clausify time	0.00	
hyper_res time	3.53	
pre_process time	10.57	
demod time	0.00	
weigh cl time	0.00	
for_sub time	2.80	
renumber time	0.92	
keep cl time	4.46	
print_cl time	0.00	
conflict time	1.17	
post_process time	2.63	

```

back demod time      0.00
back_sub time        2.57
lex_rpo time         0.00
The job finished     Wed Jun  3 13:17:14 1992

```

5.7 Theorem 7: Many-valued Sentential Calculus, CD-60

```

----- OTTER 2.2, July 1991 -----
The job began on altair.mcs.anl.gov, Wed Jun  3 13:31:24 1992
The command was "otter22".

```

```

set(hyper_res).
set(back_demod).
set(dynamic_demod_all).
clear(print_kept).
assign(pick_given_ratio,5).
assign(max_mem,20000).
set(control_memory).

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

list(sos).
2 [] P(i(x,i(y,x))).
3 [] P(i(i(x,y),i(i(y,z),i(x,z)))).
4 [] P(i(i(i(x,y),y),i(i(y,x),x))).
5 [] P(i(i(n(x),n(y)),i(y,x))).
6 [] -P(i(i(a,b),i(n(b),n(a)))). 
end_of_list.
OTTER sets dynamic_demod, because back_demod is set.
OTTER sets order_eq, because dynamic_demod is set.

Resetting weight limit to 13.
----> UNIT CONFLICT at 2184.96 sec ----> 16257 [binary,16256,6] .
Level of proof is 13, length is 24.

```

```

----- PROOF -----

```

```

1 [] -P(x) | -P(i(x,y)) | P(y).
2 [] P(i(x,i(y,x))).
3 [] P(i(i(x,y),i(i(y,z),i(x,z)))).
4 [] P(i(i(i(x,y),y),i(i(y,x),x))).
5 [] P(i(i(n(x),n(y)),i(y,x))).
6 [] -P(i(i(a,b),i(n(b),n(a)))). 
7 [hyper,2,1,2] P(i(x,i(y,i(z,y)))). 
13 [hyper,3,1,5] P(i(i(i(x,y),z),i(i(n(y),n(x)),z))).
14 [hyper,3,1,3] P(i(i(i(i(x,y),i(z,y)),u),i(i(z,x),u))).
15 [hyper,3,1,2] P(i(i(i(x,y),z),i(y,z))).
18 [hyper,4,1,7] P(i(i(i(x,i(y,x)),z),z)).
21 [hyper,15,1,5] P(i(n(x),i(x,y))).
22 [hyper,15,1,4] P(i(x,i(i(x,y),y))).
27 [hyper,21,1,3] P(i(i(i(x,y),z),i(n(x),z))).

```

```

33 [hyper,22,1,5] P(i(i(i(n(x),n(y)),i(y,x)),z),z)).
37 [hyper,22,1,3] P(i(i(i(x,y),y),z),i(x,z))).
59 [hyper,18,1,15] P(i(x,x)).
63 [hyper,59,1,22] P(i(i(i(x,x),y),y)).
238 [hyper,13,1,63] P(i(i(n(x),n(i(y,y))),x)).
284 [hyper,238,1,27] P(i(n(n(x)),x)).
320 [hyper,284,1,5] P(i(x,n(n(x))).
321 [hyper,284,1,3] P(i(i(x,y),i(n(n(x)),y))).
378 [hyper,320,1,22] P(i(i(i(x,n(n(x))),y),y)).
1651 [hyper,378,1,14] P(i(i(x,y),i(x,n(n(y))))).
1762 [hyper,33,1,14] P(i(i(x,i(n(y),n(z))),i(x,i(z,y))))).
2121 [hyper,37,1,14] P(i(i(x,i(y,z)),i(y,i(x,z)))).
3351 [hyper,1651,1,37] P(i(x,i(i(x,y),n(n(y))))).
5608 [hyper,3351,1,321] P(i(n(n(x)),i(i(x,y),n(n(y))))).
15901 [hyper,5608,1,2121] P(i(i(x,y),i(n(n(x)),n(n(y))))).
16256 [hyper,1762,1,15901] P(i(i(x,y),i(n(y),n(x)))).
16257 [binary,16256,6] .

```

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

----- statistics -----

clauses input	6
clauses given	2768
clauses generated	3214280
demod & eval rewrites	0
clauses wt,lit,sk delete	1712800
tautologies deleted	0
clauses forward subsumed	1485230
(subsumed by sos)	16917
clauses kept	16250
new demodulators	0
empty clauses	1
clauses back demodulated	0
clauses back subsumed	24
sos size	13466
Kbytes malloced	7216

----- times (seconds) -----

run time	2185.01	(run time 0 hr, 36 min, 25 sec)
system time	689.15	
input time	0.00	
clausify time	0.00	
hyper_res time	845.48	
pre_process time	1215.06	
demod time	0.00	
weigh cl time	271.12	
for_sub time	272.29	
renumber time	270.75	
keep cl time	18.23	
print_cl time	0.00	
conflict time	3.61	
post_process time	26.66	
back demod time	0.00	
back_sub time	25.92	

```

lex_rpo time      0.00
The job finished      Wed Jun  3 14:26:09 1992

```

6 Summary of OTTER Outputs for the Equality Set

6.1 Theorem EQ-1: The Commutator Theorem

```

----- OTTER 2.2, July 1991 -----
The job began on altair.mcs.anl.gov, Fri Jun  5 14:33:33 1992
The command was "otter22".

```

```

set(knuth_bendix).
set(index_for_back_demod).
set(process_input).
assign(max_mem,16000).
set(control_memory).
set(lex_rpo).
clear(print_kept).
clear(print_new_demod).
clear(print_back_demod).

lex([a,b,e,f(x,x),g(x),h(x,x)]).

lrpo_lr_status([f(x,x)]).

list(usable).
0 [] (x = x).
0 [] (f(e,x) = x).
0 [] (f(g(x),x) = e).
0 [] (f(f(x,y),z) = f(x,f(y,z))).
0 [] (h(x,y) = f(x,f(y,f(g(x),g(y))))).
end_of_list.

list(sos).
0 [] (f(x,f(x,x)) = e).
0 [] (h(h(a,b),b) != e).
end_of_list.

OTTER sets dynamic_demod_all, because knuth_bendix is set.
OTTER clears para_into_right, because knuth_bendix is set.
OTTER sets back_demod, because knuth_bendix is set.
OTTER sets para_from, because knuth_bendix is set.
OTTER sets para_into, because knuth_bendix is set.
OTTER clears para_from_right, because knuth_bendix is set.
OTTER sets dynamic_demod, because back_demod is set.
OTTER sets order_eq, because dynamic_demod is set.

-----> process usable:
** KEPT: 1 [] (x = x).
++++ cannot make into demodulator: 1 [] (x = x).
** KEPT: 2 [] (f(e,x) = x).
** KEPT: 4 [] (f(g(x),x) = e).
** KEPT: 6 [] (f(f(x,y),z) = f(x,f(y,z))).
** KEPT: 8 [] (h(x,y) = f(x,f(y,f(g(x),g(y))))).

```

```

-----> process sos:
** KEPT: 10 [] (f(x,f(x,x)) = e).
** KEPT: 12 [demod,9,9,7,7,7] (f(a,f(b,f(g(a),f(g(b),f(b,f(g(f(a,f(b,f(g(a),
g(b))))),g(b))))))) != e).

-----> done processing input.

----> UNIT CONFLICT at 1.49 sec ---> 156 [binary,155,1] .
Level of proof is 13, length is 19.

----- PROOF -----
1 [] (x = x).
3,2 [] (f(e,x) = x).
5,4 [] (f(g(x),x) = e).
7,6 [] (f(f(x,y),z) = f(x,f(y,z))).
9,8 [] (h(x,y) = f(x,f(y,f(g(x),g(y))))).
11,10 [] (f(x,f(x,x)) = e).
12 [demod,9,9,7,7,7]
(f(a,f(b,f(g(a),f(g(b),f(b,f(g(f(a,f(b,f(g(a),g(b))))),g(b))))))) != e).
13 [para_into,10,6,demod,7] (f(x,f(y,f(x,f(y,f(x,y)))))) = e).
16,15 [para_from,10,6,demod,3,7] (f(y,f(y,f(y,x))) = x).
18,17 [para_into,15,10] (f(x,f(x,e)) = f(x,x)).
21 [para_into,15,4,demod,18] (f(g(x),g(x)) = x).
24,23 [para_into,15,10] (f(x,e) = x).
28,27 [para_from,21,6] (f(g(x),f(g(x),y)) = f(x,y)).
32,31 [para_into,27,21,demod,5] (f(x,g(x)) = e).
34,33 [para_into,27,15,demod,28] (f(g(x),y) = f(x,f(x,y))).
36,35 [para_into,27,10,demod,24,34,32,24] (g(x) = f(x,x)).
37 [back_demod,12,demod,36,36,36,36,7,36,7,7,7,7,7,36,7,7,7,7,7,7,7,7,7,7,7,7,7,11,24,7,16,7,16]
(f(a,f(b,f(b,f(a,f(a,f(b,f(b,f(a,f(b,f(a,f(a,b))))))))))) != e).
43,42 [para_from,13,15,demod,24] (f(y,f(x,f(y,f(x,y)))) = f(x,x)).
45,44 [para_into,42,6,demod,7,7]
(f(x,f(y,f(z,f(x,f(y,f(z,x)))))) = f(y,f(z,f(y,z)))).
```

(f(x,f(y,f(z,f(x,f(y,f(z,x)))))) = f(y,f(z,f(y,z)))).

```

48,47 [para_from,42,6,demod,7,7,7,7]
(f(z,f(x,f(z,f(x,f(z,y)))))) = f(x,f(x,y)).
56 [para_into,47,10,demod,24] (f(x,f(y,f(x,y))) = f(y,f(y,f(x,x)))).
```

(f(y,f(x,f(y,f(x,f(y,f(x,x)))))) = f(x,y)).

```

62 [para_into,56,56,demod,7,7,16,11,24,7,7,7,7]
(f(x,f(y,f(x,f(y,f(x,f(y,f(x,x))))))) = f(y,f(x,f(y,y)))).
```

(f(x,f(y,f(y,f(x,f(y,f(x,f(x,y))))))) = f(y,f(y,f(x,f(y,y))))).

```

127 [para_from,62,47,demod,48] (f(x,f(y,f(y,x))) = f(y,f(x,f(x,y)))).
```

(f(x,f(y,f(y,f(x,f(y,f(x,f(x,y))))))) = f(y,f(y,f(x,f(y,y))))).

```

155 [back_demod,37,demod,153,45,43,11] (e != e).
156 [binary,155,1] .
```

----- end of proof -----
----- statistics -----
clauses input 7
clauses given 19
clauses generated 542
demod & eval rewrites 1945
tautologies deleted 0

```

clauses forward subsumed      453
  (subsumed by sos)          31
clauses kept                  114
new demodulators              41
empty clauses                 1
clauses back demodulated     18
clauses back subsumed         0
sos size                      79
Kbytes malloced               255

----- times (seconds) -----
run time                      1.53           (run time 0 hr, 0 min, 1 sec)
system time                    0.28
input time                     0.01
  classify time                0.00
  process input                 0.02
para_into time                 0.10
para_from time                 0.03
pre_process time               1.22
demod time                     0.55
weigh cl time                  0.00
for_sub time                   0.06
renumber time                  0.05
keep cl time                   0.31
print_cl time                  0.01
conflict time                  0.03
post_process time              0.11
back demod time                0.04
back_sub time                  0.07
lex_rpo time                   0.13
The job finished             Fri Jun  5 14:33:35 1992

```

6.2 Theorem EQ-2: Robbins Algebra, $(\exists c, c + c = c) \rightarrow \text{Boolean}$

```

----- OTTER 2.2, July 1991 -----
The job began on altair.mcs.anl.gov, Fri Jun  5 14:29:55 1992
The command was "otter22".

```

```

set(knuth_bendix).
set(index_for_back_demod).
set(process_input).
assign(max_mem,16000).
set(control_memory).
set(lex_rpo).
clear(print_kept).
clear(print_new_demod).
clear(print_back_demod).

lex([a,b,c,o(x,x),n(x)]).

lrpo_lr_status([o(x,x)]).

list(usable).

```

```

0 [] (x = x).
0 [] (o(x,y) = o(y,x)).
0 [] (o(o(x,y),z) = o(x,o(y,z))).
0 [] (n(o(n(o(x,y)),n(o(x,n(y)))))) = x).
end_of_list.

list(sos).
0 [] (o(c,c) = c).
0 [] (o(n(o(a,n(b))),n(o(n(a),n(b)))) != b).
end_of_list.
OTTER sets dynamic_demod_all, because knuth_bendix is set.
OTTER clears para_into_right, because knuth_bendix is set.
OTTER sets back_demod, because knuth_bendix is set.
OTTER sets para_from, because knuth_bendix is set.
OTTER sets para_into, because knuth_bendix is set.
OTTER clears para_from_right, because knuth_bendix is set.
OTTER sets dynamic_demod, because back_demod is set.
OTTER sets order_eq, because dynamic_demod is set.

-----> process usable:
** KEPT: 1 [] (x = x).
++++ cannot make into demodulator: 1 [] (x = x).
** KEPT: 2 [] (o(x,y) = o(y,x)).
++++ cannot make into demodulator: 2 [] (o(x,y) = o(y,x)).
** KEPT: 3 [] (o(o(x,y),z) = o(x,o(y,z))).
** KEPT: 5 [] (n(o(n(o(x,y)),n(o(x,n(y)))))) = x).

-----> process sos:
** KEPT: 7 [] (o(c,c) = c).
** KEPT: 9 [] (o(n(o(a,n(b))),n(o(n(a),n(b)))) != b).

-----> done processing input.

Resetting weight limit to 18.
Resetting weight limit to 17.
----> UNIT CONFLICT at 98.19 sec ----> 7578 [binary,7577,1] .
Level of proof is 31, length is 62.

----- PROOF -----
1 [] (x = x).
2 [] (o(x,y) = o(y,x)).
4,3 [] (o(o(x,y),z) = o(x,o(y,z))).
5 [] (n(o(n(o(x,y)),n(o(x,n(y)))))) = x).
8,7 [] (o(c,c) = c).
9 [] (o(n(o(a,n(b))),n(o(n(a),n(b)))) != b).
10 [para_from,7,5] (n(o(n(c),n(o(c,n(c)))))) = c).
13,12 [para_from,7,3] (o(c,o(c,x)) = o(c,x)).
15,14 [para_into,12,2] (o(c,o(x,c)) = o(c,x)).
16 [para_from,12,5] (n(o(n(o(c,x)),n(o(c,n(o(c,x)))))) = c).
18 [para_into,14,3] (o(c,o(x,o(y,c))) = o(c,o(x,y))).
20 [para_from,14,5] (n(o(n(o(c,x)),n(o(c,n(o(x,c)))))) = c).
23,22 [para_from,14,3,demod,4,4] (o(c,o(x,o(c,y))) = o(c,o(x,y))).
26 [para_into,10,2] (n(o(n(o(c,n(c)))),n(c))) = c).

```

```

28 [para_from,10,5] ( $n(o(n(o(n(c),o(c,n(c)))),c)) = n(c))$ .
42 [para_from,26,5] ( $n(o(n(o(n(o(c,n(c)),c),c)) = n(o(c,n(c))))$ .
45,44 [para_from,26,5] ( $n(o(n(o(x,o(n(o(c,n(c)),n(c)))),n(o(x,c)))) = x)$ .
56 [para_into,18,2,demod,4,23] ( $o(c,o(x,y)) = o(c,o(y,x)))$ .
57 [para_into,18,2,demod,4,4,8] ( $o(x,o(y,c)) = o(c,o(x,y)))$ .
64 [para_into,56,2,demod,4] ( $o(x,o(y,c)) = o(c,o(y,x)))$ .
68 [para_from,56,2,demod,4] ( $o(c,o(x,y)) = o(y,o(x,c)))$ .
70 [para_into,57,2] ( $o(x,o(c,y)) = o(c,o(x,y)))$ .
71 [para_into,57,2,demod,4] ( $o(x,o(c,y)) = o(c,o(y,x)))$ .
72 [para_from,57,5] ( $n(o(n(o(c,o(x,y))),n(o(x,n(o(y,c)))))) = x$ ).
76 [para_from,57,2,demod,4] ( $o(c,o(x,y)) = o(y,o(c,x)))$ .
82 [para_from,64,5] ( $n(o(n(o(c,o(x,y))),n(o(y,n(o(x,c)))))) = y$ ).
86 [para_from,64,2,demod,4] ( $o(c,o(x,y)) = o(x,o(c,y)))$ .
105 [para_into,76,68] ( $o(x,o(y,c)) = o(x,o(c,y)))$ .
114 [para_into,86,76] ( $o(x,o(c,y)) = o(y,o(c,x)))$ .
163 [para_into,9,2] ( $o(n(o(n(b),a)),n(o(n(a),n(b)))) != b$ ).
166 [para_into,28,71] ( $n(o(n(o(c,o(n(c),n(c)))),c)) = n(c))$ .
168 [para_into,28,2] ( $n(o(c,n(o(n(c),o(c,n(c)))))) = n(c))$ .
180 [para_into,163,2] ( $o(n(o(n(b),a)),n(o(n(b),n(a)))) != b$ ).
187 [para_from,166,5] ( $n(o(n(c),n(o(n(o(c,o(n(c),n(c)))),n(c)))))) = n(o(c,o(n(c),n(c))))$ ).
190,189 [para_from,168,5,demod,23] ( $n(o(n(o(c,o(n(c),n(c)))),n(c))) = c$ ).
196,195 [back_demod,187,demod,190] ( $n(o(c,o(n(c),n(c)))) = n(o(n(c),c))$ ).
201 [back_demod,166,demod,196] ( $n(o(n(o(n(c),c)),c)) = n(c))$ .
210,209 [para_into,201,2] ( $n(o(n(o(c,n(c))),c)) = n(c))$ .
212,211 [back_demod,42,demod,210] ( $n(o(n(c),c)) = n(o(c,n(c))))$ .
219 [back_demod,195,demod,212] ( $n(o(c,o(n(c),n(c)))) = n(o(c,n(c))))$ .
230 [para_into,219,68] ( $n(o(n(c),o(n(c),c))) = n(o(c,n(c))))$ .
240 [para_into,16,114,demod,8,13] ( $n(o(n(o(x,c)),n(o(c,n(o(c,x)))))) = c$ ).
276 [para_into,20,2] ( $n(o(n(o(c,x)),n(o(n(o(x,c)),c)))) = c$ ).
1150 [para_into,240,2] ( $n(o(n(o(x,c)),n(o(n(o(c,x)),c)))) = c$ ).
2492 [para_into,72,105,demod,13,8] ( $n(o(n(o(c,x)),n(o(x,n(c)))))) = x$ ).
2517,2516 [para_into,72,276] ( $n(o(n(o(c,o(n(o(c,x)),n(o(x,c))))),c)) = n(o(c,x))$ ).
2590 [para_into,2492,114,demod,8,4] ( $n(o(n(o(x,c)),n(o(c,o(x,n(c)))))) = o(c,x))$ .
2636 [para_into,2492,2] ( $n(o(n(o(c,x)),n(o(n(c),x)))) = x$ ).
2642 [para_into,2492,2] ( $n(o(n(o(x,n(c))),n(o(c,x)))) = x$ ).
2788 [para_into,2636,230,demod,15] ( $n(o(n(o(c,n(c))),n(o(c,n(c)))))) = o(n(c),c))$ .
3454 [para_into,82,1150,demod,2517] ( $n(o(c,x)) = n(o(x,c))$ ).
3508 [para_from,3454,5] ( $n(o(n(o(c,x)),n(o(n(x),c)))) = c$ ).
3777 [para_into,3508,2] ( $n(o(n(o(n(x),c)),n(o(c,x)))) = c$ ).
3912,3911 [para_into,3777,26,demod,8] ( $n(o(n(c),n(o(c,o(n(o(c,n(c)),n(c)))))) = c$ ).
5861 [para_into,2590,209,demod,3912] ( $o(c,n(o(c,n(c)))) = c$ ).
5928 [para_from,5861,70] ( $o(c,o(x,n(o(c,n(c)))))) = o(x,c))$ .
5983,5982 [para_from,5928,2642,demod,4,45] ( $o(x,n(o(c,n(c)))) = x$ ).
6005,6004 [back_demod,2788,demod,5983] ( $n(n(o(c,n(c)))) = o(n(c),c))$ .
6015,6014 [para_into,5982,2] ( $o(n(o(c,n(c))),x) = x$ ).
6039,6038 [para_from,5982,5,demod,6005] ( $n(o(n(x),n(o(x,o(n(c),c)))))) = x$ ).
6040 [para_from,6014,5,demod,6015] ( $n(o(n(x),n(n(x)))) = n(o(c,n(c))))$ .
7006 [para_into,6040,6038,demod,6039] ( $n(o(x,n(x))) = n(o(c,n(c))))$ .
7023 [para_into,7006,2] ( $n(o(n(x),x)) = n(o(c,n(c))))$ .
7066 [para_from,7006,5,demod,6015] ( $n(n(o(x,n(n(x)))),x)) = x$ ).
7147,7146 [para_into,7066,2] ( $n(n(o(n(n(x)),x))) = x$ ).
7416,7415 [para_from,7023,5,demod,5983,7147] ( $n(n(x)) = x$ ).
7576,7575 [para_into,7415,5] ( $o(n(o(x,y)),n(o(x,n(y)))) = n(x))$ .
7577 [back_demod,180,demod,7576,7416] ( $b != b$ ).

```

```

7578 [binary,7577,1] .

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


----- statistics -----
clauses input          6
clauses given          249
clauses generated      50001
demod & eval rewrites  122587
clauses wt,lit,sk delete 901
tautologies deleted    0
clauses forward subsumed 45667
  (subsumed by sos)   10963
clauses kept           4548
new demodulators       3029
empty clauses          1
clauses back demodulated 1109
clauses back subsumed   3
sos size               3290
Kbytes malloced        5652

----- times (seconds) -----
run time                98.31          (run time 0 hr, 1 min, 38 sec)
system time              18.22
input time               0.01
  classify time          0.00
  process input           0.03
para_into time           6.66
para_from time           4.05
pre_process time         63.38
  demod time             24.30
  weigh cl time          0.19
  for_sub time            7.08
  renumber time           3.83
  keep cl time            8.73
  print_cl time           0.00
  conflict time           0.63
post_process time        22.50
  back demod time        16.85
  back_sub time           5.52
lex_rpo time             9.34
The job finished         Fri Jun  5 14:31:54 1992

```

6.3 Theorem EQ-3: On Ternary Boolean Algebra

```

----- OTTER 2.2, July 1991 -----
The job began on altair.mcs.anl.gov, Fri Jun  5 07:39:41 1992
The command was "otter22".

```

```

set(knuth_bendix).
set(index_for_back_demod).
set(process_input).
assign(max_mem,16000).

```

```

set(control_memory).
set(lex_rpo).
clear(print_kept).
clear(print_new_demod).
clear(print_back_demod).

lex([a,b,c,f(x,x,x),g(x)]).

lrpo_lr_status([f(x,x,x)]).

list(usable).
0 [] (x = x).
end_of_list.

list(sos).
0 [] (f(f(v,w,x),y,f(v,w,z)) = f(v,w,f(x,y,z))).
0 [] (f(y,x,x) = x).
0 [] (f(x,x,y) = x).
0 [] (f(g(y),y,x) = x).
0 [] (f(a,g(a),b) != b).
end_of_list.

OTTER sets dynamic_demod_all, because knuth_bendix is set.
OTTER clears para_into_right, because knuth_bendix is set.
OTTER sets back_demod, because knuth_bendix is set.
OTTER sets para_from, because knuth_bendix is set.
OTTER sets para_into, because knuth_bendix is set.
OTTER clears para_from_right, because knuth_bendix is set.
OTTER sets dynamic_demod, because back_demod is set.
OTTER sets order_eq, because dynamic_demod is set.

-----> process usable:
** KEPT: 1 [] (x = x).
++++ cannot make into demodulator: 1 [] (x = x).

-----> process sos:
** KEPT: 2 [] (f(f(x,y,z),u,f(x,y,v)) = f(x,y,f(z,u,v))).
** KEPT: 4 [] (f(x,y,y) = y).
** KEPT: 6 [] (f(x,x,y) = x).
** KEPT: 8 [] (f(g(x),x,y) = y).
** KEPT: 10 [] (f(a,g(a),b) != b).

-----> done processing input.

----> UNIT CONFLICT at 16.78 sec ----> 1950 [binary,1948,10] .
Level of proof is 11, length is 19.

----- PROOF -----
3,2 [] (f(f(x,y,z),u,f(x,y,v)) = f(x,y,f(z,u,v))).
5,4 [] (f(x,y,y) = y).
7,6 [] (f(x,x,y) = x).
9,8 [] (f(g(x),x,y) = y).
10 [] (f(a,g(a),b) != b).
12,11 [para_into,2,6,demod,7,7] (f(x,y,x) = x).

```

```

16 [para_into,2,4] (f(f(x,y,z),u,y) = f(x,y,f(z,u,y))).  

21,20 [para_into,2,6] (f(x,y,f(z,f(x,y,z),u)) = f(x,y,z)).  

24 [para_from,11,2] (f(f(x,y,z),u,x) = f(x,y,f(z,u,x))).  

50 [para_into,16,11] (f(x,z,f(x,y,z)) = f(x,y,z)).  

53,52 [para_into,16,2] (f(f(x,y,f(z,u,v)),w,u) = f(f(x,y,z),u,f(f(x,y,v),w,u))).  

82 [para_into,20,4] (f(x,y,f(x,y,z)) = f(x,y,z)).  

90 [para_from,20,50,demod,21] (f(x,f(y,f(x,z,y),u),f(x,z,y)) = f(x,z,y)).  

102 [para_from,82,2,demod,3] (f(x,y,f(z,u,f(x,y,v))) = f(x,y,f(z,u,v))).  

351,350 [para_from,90,20,demod,5] (f(x,f(y,z,x),y) = f(y,z,x)).  

386,385 [para_from,350,24] (f(f(x,y,z),u,z) = f(z,f(x,y,z),f(x,u,z))).  

445 [para_into,102,20,demod,5] (f(x,f(y,z,x),f(y,z,u)) = f(y,z,x)).  

506,505 [para_into,445,50] (f(x,f(y,z,x),f(y,u,z)) = f(y,z,x)).  

524,523 [para_into,445,4] (f(x,f(y,z,x),z) = f(y,z,x)).  

781 [para_from,505,350,demod,506] (f(f(x,y,z),f(x,z,u),u) = f(x,z,u)).  

1301 [para_into,781,523,demod,53,12,524,524] (f(f(x,y,z),u,f(z,u,x)) = f(z,u,x)).  

1734 [para_into,1301,523] (f(f(x,y,z),u,f(y,u,z)) = f(y,u,z)).  

1855 [para_into,1734,8,demod,386,9] (f(z,f(x,g(y),z),f(x,y,z)) = z).  

1948 [para_into,1855,6,demod,351] (f(y,g(y),x) = x).  

1950 [binary,1948,10] .

```

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

----- statistics -----

clauses input	6
clauses given	47
clauses generated	3945
demod & eval rewrites	6838
tautologies deleted	0
clauses forward subsumed	3103
(subsumed by sos)	19
clauses kept	1030
new demodulators	919
empty clauses	1
clauses back demodulated	182
clauses back subsumed	3
sos size	798
Kbytes malloced	1564

----- times (seconds) -----

run time	16.81	(run time 0 hr, 0 min, 16 sec)
system time	1.85	
input time	0.01	
clausify time	0.00	
process input	0.02	
para_into time	0.50	
para_from time	0.43	
pre_process time	11.37	
demod time	6.77	
weigh cl time	0.00	
for_sub time	0.92	
renumber time	0.28	
keep cl time	1.41	
print_cl time	0.00	
conflict time	0.17	

```

post_process time      4.32
  back demod time    3.01
  back_sub time     1.30
lex_rpo time        0.40
The job finished      Fri Jun  5 07:40:00 1992

```

6.4 Theorem EQ-4: Group Theory Single Axiom

```

----- OTTER 2.2, July 1991 -----
The job began on altair.mcs.anl.gov, Fri Jun  5 07:36:59 1992
The command was "otter22".

```

```

set(knuth_bendix).
set(index_for_back_demod).
set(process_input).
assign(max_mem,16000).
set(control_memory).
set(lex_rpo).
clear(print_kept).
clear(print_new_demod).
clear(print_back_demod).

lex([a,b,c,f(x,x),i(x)]).

lrpo_lr_status([f(x,x)]).

list(usable).
0 [] (x = x).
end_of_list.

list(sos).
0 [] (f(x,i(f(f(i(f(i(y),f(i(x),w))),z),i(f(y,z)))))) = w).
0 [] (f(a,f(b,c)) != f(f(a,b),c)).
end_of_list.

OTTER sets dynamic_demod_all, because knuth_bendix is set.
OTTER clears para_into_right, because knuth_bendix is set.
OTTER sets back_demod, because knuth_bendix is set.
OTTER sets para_from, because knuth_bendix is set.
OTTER sets para_into, because knuth_bendix is set.
OTTER clears para_from_right, because knuth_bendix is set.
OTTER sets dynamic_demod, because back_demod is set.
OTTER sets order_eq, because dynamic_demod is set.

-----> process usable:
** KEPT: 1 [] (x = x).
++++ cannot make into demodulator: 1 [] (x = x).

-----> process sos:
** KEPT: 2 [] (f(x,i(f(f(i(f(i(y),f(i(x),z))),u),i(f(y,u)))))) = z).
** KEPT: 4 [] (f(f(a,b),c) != f(a,f(b,c))).

-----> done processing input.

```

----> UNIT CONFLICT at 44.12 sec ----> 4092 [binary,4090,4] .
Level of proof is 50, length is 92.

----- PROOF -----

```

3,2 [] (f(x,i(f(f(i(f(i(y),f(i(x),z))),u),i(f(y,u)))))) = z).
4 [] (f(f(a,b),c) != f(a,f(b,c))).
5 [para_intro,2,2] (f(x,i(f(f(i(f(i(y),z)),u),i(f(y,u)))))) = i(f(f(i(f(i(v),
f(i(i(x)),z))),w),i(f(v,w))))).
12 [para_intro,5,5] (f(x,i(f(f(i(f(i(y),f(i(i(f(i(z),u)))),v))),w),
i(f(y,w)))),i(f(z,i(f(f(i(v6),v),v7),i(f(v6,v7))))))) = i(f(f(i(f(i(v8),
f(i(i(x)),u))),v9),i(f(v8,v9)))).
19,18 [para_intro,5,2] (i(f(f(i(f(i(y),f(i(i(z),f(i(z),x)))),u),i(f(y,u)))))) = x).
29 [para_intro,18,18,demod,19] (i(f(f(i(f(i(x),f(i(y),f(y,w)))),v6),i(f(x,v6)))))) = w).
48 [para_intro,29,5] (i(f(i(f(f(i(f(i(x),f(i(i(f(i(y),f(i(z),f(z,u)))),v)))),v)),
w),i(f(x,w)))),i(f(y,i(f(f(i(f(i(v6),v),v7),i(f(v6,v7)))))))))) = u).
50 [para_intro,29,2] (i(f(x,i(f(y,i(f(f(i(f(i(z),f(i(i(f(i(y),f(i(u),f(u,v)))),v)),
x))),w),i(f(z,w))))))) = v).
60 [para_from,29,5] (i(f(f(i(f(i(z),f(i(i(x),f(i(u),f(u,y)))),v),i(f(z,v)))))) = f(x,y)).
81 [para_intro,60,5] (i(f(i(f(f(i(f(i(x),f(i(i(f(i(y),f(i(i(z),f(i(u),
f(u,v)))),w))),v6),i(f(x,v6)))),i(f(y,i(f(f(i(f(i(v7),w),
v8),i(f(v7,v8)))))))))) = f(z,v)).
98,97 [para_from,60,5] (i(f(f(i(f(i(u),f(i(i(x),f(i(i(y),f(i(v),f(v,z)))),v)),
w),i(f(u,w)))))) = f(x,f(y,z))).
99 [para_from,60,2] (f(i(x),f(x,y)) = f(i(z),f(z,y))).
114 [para_intro,99,99] (f(i(i(x)),f(i(y),f(y,z))) = f(i(u),f(u,f(x,z)))).
116,115 [para_intro,99,5,demod,3] (f(i(v),f(v,i(f(f(i(f(i(w),z),v6),i(f(w,v6))))))) = z).
125 [para_from,99,29] (i(f(f(i(f(i(x),f(x,f(y,z)))),u),i(f(i(y),u)))))) = z).
136 [para_from,99,2] (f(x,i(f(f(i(f(i(y),f(y,z)),u),i(f(i(x),u)))))) = z).
146 [para_from,114,60,demod,98] (f(y,f(i(z),f(z,v))) = f(y,f(i(v6),f(v6,v)))).
158 [para_intro,146,99] (f(x,f(i(i(y)),f(i(z),f(z,u)))) = f(x,f(i(v),f(v,f(y,u))))).
160 [para_from,146,99] (f(i(x),f(x,f(i(y),f(y,z)))) = f(i(u),f(u,f(i(v),f(v,z))))).
210 [para_intro,115,99] (f(i(x),f(x,i(f(f(i(f(i(y),f(y,z)),u),i(f(v,u))))))) = f(v,z)).
221,220 [para_intro,115,99] (f(i(x),f(x,i(f(f(i(y),f(y,z)),i(f(u,f(f(i(u),v),z))))))) = v).
224 [para_intro,115,2] (f(i(x),f(x,i(f(y,i(f(z,i(f(f(i(f(i(u),f(i(i(f(i(z),v),
y))),w),i(f(u,w)))))))))) = v).
300,299 [para_intro,125,99] (i(f(f(i(f(i(x),f(x,f(y,z)))),f(y,u),i(f(i(v),f(v,u)))))) = z).
355 [para_intro,136,99] (f(x,i(f(f(i(y),f(y,z)),i(f(i(x),f(f(i(u),f(u,v)),z)))))) = v).
1301 [para_intro,355,220,demod,221] (f(f(i(x),f(x,y)),i(f(v6,i(v6)))))) = y).
1410 [para_from,1301,210] (f(i(f(i(x),f(x,y))),y) = f(i(f(i(z),f(z,u))),u)).
1434 [para_from,1301,220] (f(i(x),f(x,i(f(f(i(f(i(y),f(y,z)),z),i(f(u,
f(f(i(u),v),i(f(w,i(w)))))))))) = v).
1436 [para_from,1301,114] (f(i(i(x)),f(i(f(i(y),f(y,z)),z)) =
f(i(u),f(u,f(x,i(f(v,i(v)))))))..
1484 [para_from,1410,355] (f(f(i(x),f(x,f(f(i(y),f(y,z)),u))),i(f(f(i(v),
f(v,u)),i(f(i(f(i(w),f(w,v6))),v6)))))) = z).
1489,1488 [para_from,1410,220,demod,221] (f(i(i(v)),f(i(f(i(w),f(w,v6))),v6)) = v).
1493,1492 [para_from,1410,210] (f(i(x),f(x,i(f(f(i(f(i(y),f(y,z)),z),i(f(u,v))))))) = f(u,v)).
1541 [back_demod,1436,demod,1489] (f(i(u),f(u,f(x,i(f(v,i(v))))))) = x).
1545,1544 [back_demod,1434,demod,1493] (f(u,f(f(i(u),v),i(f(w,i(w)))))) = v).
1591 [para_from,1541,1301] (f(x,i(f(y,i(y)))) = f(x,i(f(z,i(z))))).
1675 [para_intro,1544,5] (f(x,f(i(f(f(i(f(i(y),f(i(i(i(x)),z)))),u),i(f(y,u)))),i(f(v,i(v)))))) = i(f(f(i(f(i(w),z),v6),i(f(w,v6))))).
1802 [para_from,1488,210] (f(i(x),f(x,i(f(f(i(f(i(y),f(y,z)),f(i(f(i(u),

```

```

f(u,v))),v)),i(w)))) = f(i(i(w)),z)).
1857 [para_from,1591,1488] (f(i(i(x)),f(i(f(i(y),f(y,i(f(z,i(z)))))),i(f(u,i(u))))) = x).
1884 [para_from,1591,115,demod,116] (i(f(z,i(z))) = i(f(v,i(v)))).
1948 [para_from,1884,1488,demod,1489] (f(x,i(x)) = f(u,i(u))).
1969 [para_from,1884,99] (f(i(f(x,i(x))),f(f(y,i(y)),z)) = f(i(u),f(u,z))).
2010 [para_from,1948,1544] (f(x,f(y,i(y))) = i(i(x))).
2107 [para_into,2010,99] (f(i(x),f(x,i(y))) = i(i(i(y))).
2116 [para_from,2010,299] (i(f(i(i(i(f(i(x),f(x,f(y,z)))))),i(f(i(u),f(u,i(y)))))) = z).
2131 [para_from,2010,210] (f(i(x),f(x,i(f(i(i(f(i(y),f(y,z)))))),i(f(u,f(v,i(v))))))) = f(u,z)).
2368 [para_into,2107,299,demod,300] (f(i(x),f(x,y)) = i(i(y))).
2373 [para_into,2368,1884] (f(i(f(x,i(x))),f(f(y,i(y)),z)) = i(i(z))).
2381,2380 [para_into,2368,2368] (i(i(f(x,y))) = f(i(i(x)),i(i(y))).
2401 [para_into,2368,158,demod,2381,2381] (f(i(x),f(x,f(i(y),f(y,f(z,u)))))) =
f(i(i(i(i(z)))),f(i(i(i(v))),f(i(i(v)),i(i(u))))).
2438 [back_demod,2131,demod,2381,2381] (f(i(x),f(x,i(f(i(f(i(i(i(y))),f(i(i(y),
i(i(z))))),i(f(u,f(v,i(v)))))) = f(u,z)).
2446 [back_demod,2116,demod,2381,2381,2381] (i(f(i(f(i(i(i(x))),f(i(i(x)),f(i(i(y),
i(i(z))))),i(f(i(u),f(u,i(y)))))) = z).
2587 [back_demod,224,demod,2381] (f(i(x),f(x,i(f(y,i(f(z,i(f(f(i(f(i(u),f(f(i(i(i(z)),
i(i(v))),y)),w),i(f(u,w)))))))) = v).
2598 [back_demod,81,demod,2381,2381,2381] (i(f(i(f(f(i(f(i(x),f(i(f(i(i(i(y)),
f(i(i(i(i(z))),f(i(i(i(u)),f(i(i(u)),i(i(v))))))),w))),v6),i(f(x,v6))),i(f(y,i(f(f(i(f(i(v7),w),v8),i(f(v7,v8))))))) = f(z,v)).
2604 [back_demod,50,demod,2381,2381,2381] (i(f(x,i(f(y,i(f(f(i(f(i(z),f(f(i(i(i(y)),
f(i(i(i(i(u))),f(i(i(u)),i(i(v))))),x))),w),i(f(z,w))))))) = v).
2606 [back_demod,48,demod,2381,2381,2381] (i(f(i(f(f(i(f(i(x),f(i(f(i(i(i(y)),
f(i(i(i(z)),f(i(i(z)),i(i(u))))),v))),w),i(f(x,w))),i(f(y,
i(f(f(i(f(i(v6),v),v7),i(f(v6,v7))))))) = u).
2617 [back_demod,12,demod,2381] (f(x,i(f(i(f(f(i(f(i(y),f(i(f(i(i(i(z)),i(i(u)),
v))),w),i(f(y,w))),i(f(z,i(f(f(i(f(i(v6),v),v7),i(f(v6,v7))))))),i(f(f(i(f(i(v8),f(i(i(x)),u))),v9),i(f(v8,v9))))).
2621 [para_from,2368,1488] (i(i(x)) = f(i(y),f(y,x))).
2633 [para_from,2368,210] (f(i(x),f(x,i(f(i(i(y)),i(f(z,f(f(i(u),f(u,v)),y))))))) = f(z,v)).
2647 [para_from,2368,1301] (f(i(i(x)),i(f(y,i(y)))) = x).
2706 [para_from,2368,160] (f(i(x),f(x,f(i(i(y)),i(i(z)))))) = f(i(u),f(u,f(i(v),f(v,f(y,z))))).
2709 [para_from,2368,29] (i(f(f(i(f(i(x),i(i(y)))),z),i(f(x,z)))) = y).
2802,2801 [para_from,2621,1488] (f(i(f(i(x),f(x,y))),f(i(f(i(z),f(z,u))),u)) = i(y)).
2866,2865 [back_demod,1802,demod,2802] (f(i(x),f(x,i(f(i(z),i(w)))))) = f(i(i(w)),z)).
2873 [back_demod,2633,demod,2866,2381,2381,2381] (f(f(i(i(z)),f(f(i(i(u)),
f(i(i(u)),i(i(v)))),i(i(y)))),i(y)) = f(z,v)).
2879 [back_demod,2438,demod,2866,2381,2381] (f(f(i(i(u)),f(i(i(v)),i(i(i(v)))),
f(i(i(i(y))),f(i(i(y)),i(i(z)))))) = f(u,z)).
2884,2883 [para_into,2647,2621] (f(i(f(i(x),f(x,y))),i(f(z,i(z)))) = i(y)).
2921,2920 [back_demod,1857,demod,2884,2381] (f(i(i(x)),f(i(i(z)),i(i(i(z)))))) = x).
2929,2928 [back_demod,2879,demod,2921] (f(x,f(i(i(i(z))),f(i(i(z)),i(i(u)))))) = f(x,u)).
2938,2937 [back_demod,2606,demod,2929] (i(f(i(f(f(i(f(i(x),f(i(f(i(i(i(y)),u)),v))),w),
i(f(x,w))),i(f(y,i(f(f(i(f(i(v6),v),v7),i(f(v6,v7)))))))) = u).
2940,2939 [back_demod,2604,demod,2929] (i(f(x,i(f(y,i(f(f(i(f(i(z),f(f(i(i(i(y)),
v),x))),w),i(f(z,w))))))) = v).
2944,2943 [back_demod,2598,demod,2929,2938] (f(i(i(i(i(i(z)))),v) = f(z,v)).
2978,2977 [back_demod,2401,demod,2929,2944] (f(i(x),f(x,f(i(y),f(y,f(z,u)))))) = f(z,u)).
2985,2984 [back_demod,2617,demod,2938] (i(f(f(i(f(i(v8),f(i(i(x)),u))),v9),i(f(v8,v9)))) =
f(x,i(i(u))).
2987,2986 [back_demod,2587,demod,2940] (f(i(x),f(x,i(i(v)))) = v).

```

```

3017,3016 [back_demod,2706,demod,2978] (f(i(x),f(x,f(i(i(y)),i(i(z))))) = f(y,z)).
3114,3113 [back_demod,1675,demod,2985,1545] (i(f(f(i(f(i(w),z)),v6),i(f(w,v6)))) = i(i(z))).
3204 [back_demod,2873,demod,2987] (f(f(i(i(x)),f(z,i(i(u)))),i(u)) = f(x,z)).
3266 [back_demod,2446,demod,3017] (i(f(i(f(y,z)),i(f(i(u),f(u,i(y)))))) = z).
3344,3343 [back_demod,2709,demod,3114] (i(i(i(i(y)))) = y).
3706,3705 [para_into,3343,2621] (i(f(i(x),f(x,i(y)))) = y).
3711 [back_demod,3266,demod,3706] (i(f(i(f(x,y)),x)) = y).
3752 [para_from,3343,1591] (f(x,i(f(i(i(y))),y))) = f(x,i(f(z,i(z))))..
3759 [para_into,3711,1544] (i(f(i(x),y)) = f(f(i(y),x),i(f(z,i(z))))).
3773,3772 [para_into,3711,2010,demod,2381,2381,2381,2381,3344,3344,3344] (f(f(x,i(x)),y) = y).
3814 [back_demod,2373,demod,3773] (f(i(f(x,i(x))),z) = i(i(z))).
3825 [back_demod,1969,demod,3773] (f(i(f(x,i(x))),z) = f(i(u),f(u,z))).
3896,3895 [para_into,3772,1884,demod,3773] (f(i(f(y,i(y))),z) = z).
3900,3899 [para_into,3772,1544,demod,3896] (f(x,i(f(z,i(z)))) = x).
3914,3913 [back_demod,3825,demod,3896] (f(i(z),f(z,y)) = y).
3920,3919 [back_demod,3814,demod,3896] (i(i(y)) = y).
3935,3934 [back_demod,3759,demod,3900] (i(f(i(x),y)) = f(i(y),x)).
3937,3936 [back_demod,3752,demod,3920,3935,3900] (f(x,f(i(y),y)) = x).
4029 [back_demod,3204,demod,3920,3920] (f(f(x,f(y,z)),i(z)) = f(x,y)).
4042 [back_demod,1484,demod,3914,3914,3914,3914,3935,3937] (f(f(z,u),i(u)) = z).
4090 [para_into,4029,4042,demod,3920] (f(f(x,y),z) = f(x,f(y,z))).
4092 [binary,4090,4] .

```

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

----- statistics -----

clauses input	3
clauses given	57
clauses generated	3417
demod & eval rewrites	9814
tautologies deleted	0
clauses forward subsumed	3327
(subsumed by sos)	503
clauses kept	2507
new demodulators	1584
empty clauses	1
clauses back demodulated	2414
clauses back subsumed	61
sos size	15
Kbytes malloced	4470

----- times (seconds) -----

run time	89.93	(run time 0 hr, 1 min, 29 sec)
system time	3.46	
input time	0.01	
clausify time	0.00	
process input	0.01	
para_into time	0.69	
para_from time	0.83	
pre_process time	71.63	
demod time	7.86	
weigh cl time	0.00	
for_sub time	1.73	
renumber time	0.84	

```

keep cl time      9.14
print_cl time    0.00
conflict time    0.32
post_process time 16.45
  back demod time 14.77
  back_sub time   1.43
lex_rpo time     2.54
The job finished      Fri Jun  5 07:38:34 1992

```

6.5 Theorem EQ-5: On Wajsberg Algebra

----- OTTER 2.2, July 1991 -----
 The job began on altair.mcs.anl.gov, Thu Jun 4 17:31:43 1992
 The command was "otter22".

```

set(knuth_bendix).
set(index_for_back_demod).
set(process_input).
assign(max_mem,16000).
set(control_memory).
set(lex_rpo).
clear(print_kept).
clear(print_new_demod).
clear(print_back_demod).

lex([a,b,T,i(x,x),n(x)]).

lrpo_lr_status([i(x,x)]).

list(usable).
0 [] (x = x).
end_of_list.

list(sos).
0 [] (i(T,x) = x).
0 [] (i(i(x,y),i(i(y,z),i(x,z))) = T).
0 [] (i(i(x,y),y) = i(i(y,x),x)).
0 [] (i(i(n(x),n(y)),i(y,x)) = T).
0 [] (i(i(i(a,b),i(b,a)),i(b,a)) != T).
end_of_list.

OTTER sets dynamic_demod_all, because knuth_bendix is set.
OTTER clears para_into_right, because knuth_bendix is set.
OTTER sets back_demod, because knuth_bendix is set.
OTTER sets para_from, because knuth_bendix is set.
OTTER sets para_into, because knuth_bendix is set.
OTTER clears para_from_right, because knuth_bendix is set.
OTTER sets dynamic_demod, because back_demod is set.
OTTER sets order_eq, because dynamic_demod is set.

-----> process usable:
** KEPT: 1 [] (x = x).
**** cannot make into demodulator: 1 [] (x = x).

```

```

-----> process sos:
** KEPT: 2 [] (i(T,x) = x).
** KEPT: 4 [] (i(i(x,y),i(i(y,z),i(x,z))) = T).
** KEPT: 6 [] (i(i(x,y),y) = i(i(y,x),x)).
++++ cannot make into demodulator: 6 [] (i(i(x,y),y) = i(i(y,x),x)).
** KEPT: 7 [] (i(i(n(x),n(y)),i(y,x)) = T).
** KEPT: 9 [] (i(i(i(a,b),i(b,a)),i(b,a)) != T).

-----> done processing input.

```

Resetting weight limit to 15.

---> UNIT CONFLICT at 2248.86 sec ---> 11462 [binary,11460,11350] .
Level of proof is 37, length is 85.

----- PROOF -----

```

3,2 [] (i(T,x) = x).
4 [] (i(i(x,y),i(i(y,z),i(x,z))) = T).
6 [] (i(i(x,y),y) = i(i(y,x),x)).
7 [] (i(i(n(x),n(y)),i(y,x)) = T).
9 [] (i(i(i(a,b),i(b,a)),i(b,a)) != T).
10 [para_into,6,6] (i(i(i(x,y),y),x) = i(i(x,i(y,x)),i(y,x))).
12 [para_into,6,2] (i(i(x,T),T) = i(x,x)).
21 [para_into,7,2] (i(i(n(x),n(T)),x) = T).
23 [para_from,7,6,demod,3] (i(i(i(x,y),i(n(y),n(x))),i(n(y),n(x))) = i(x,y)).
25 [para_into,21,6] (i(i(n(T),n(n(T))),n(n(T))) = T).
31 [para_from,21,6,demod,3] (i(i(x,i(n(x),n(T))),i(n(x),n(T))) = x).
35 [para_from,25,6,demod,3] (i(i(n(n(T)),i(n(T),n(n(T)))),i(n(T),n(n(T)))) = n(n(T))).
49 [para_into,4,2,demod,3] (i(x,i(i(x,y),y)) = T).
63 [para_into,4,2] (i(i(x,T),i(y,i(x,y))) = T).
92,91 [para_into,49,2,demod,3] (i(x,x) = T).
98,97 [para_into,49,12,demod,92] (i(x,T) = T).
102,101 [back_demod,63,demod,98,3] (i(y,i(x,y)) = T).
103 [back_demod,35,demod,102,3] (i(n(T),n(n(T))) = n(n(T))).
106,105 [back_demod,10,demod,102,3] (i(i(i(x,y),y),x) = i(y,x)).
111 [para_from,101,4,demod,3] (i(i(x,y),i(x,i(z,y))) = T).
113 [para_from,101,6,demod,3] (i(i(i(x,y),y),y) = i(x,y)).
115 [para_from,101,4,demod,3] (i(i(i(x,y),z),i(y,z)) = T).
151 [para_into,111,49,demod,3] (i(x,i(y,i(i(x,z),z))) = T).
194,193 [para_into,115,21,demod,3] (i(n(T),x) = T).
195 [para_into,115,7,demod,3] (i(n(x),i(x,y)) = T).
210,209 [back_demod,103,demod,194] (n(n(T)) = T).
219 [para_from,193,4,demod,3] (i(i(x,n(T)),i(x,y)) = T).
222,221 [para_from,193,6,demod,3] (i(i(x,n(T)),n(T)) = x).
229 [para_from,195,4,demod,3] (i(i(x,n(y)),i(x,i(y,z))) = T).
234,233 [para_from,195,6,demod,3] (i(i(i(x,y),n(x)),n(x)) = i(x,y)).
237 [para_from,221,7,demod,210,3] (i(n(i(x,n(T))),x) = T).
248,247 [para_from,237,7,demod,3] (i(x,i(n(x),n(T))) = T).
258,257 [back_demod,31,demod,248,3] (i(n(x),n(T)) = x).
259 [para_from,257,221] (i(x,n(T)) = n(x)).
264 [para_from,257,115] (i(i(i(x,n(y)),n(T)),y) = T).
272 [para_from,257,4,demod,194,3] (i(x,i(n(x),y)) = T).
275,274 [para_into,259,257] (n(n(x)) = x).
276 [para_into,259,221] (n(i(x,n(T))) = x).

```

```

278 [para_from,259,113,demod,222] (n(x) = i(x,n(T))).  

288,287 [para_from,274,7] (i(i(n(x),y),i(n(y),x)) = T).  

289 [para_from,274,7] (i(i(x,n(y)),i(y,n(x))) = T).  

291 [para_from,278,7] (i(i(n(x),i(y,n(T))),i(y,x)) = T).  

322,321 [para_from,219,6,demod,3] (i(i(i(x,y),i(x,n(T))),i(x,n(T))) = i(x,y)).  

377 [para_into,151,6] (i(x,i(i(i(x,y),y),z),z)) = T).  

417 [para_into,264,259] (i(n(i(x,n(y))),y) = T).  

429 [para_into,417,274] (i(n(i(x,y)),n(y)) = T).  

441 [para_into,429,6] (i(n(i(i(x,y),y)),n(x)) = T).  

474,473 [para_into,287,274] (i(i(x,y),i(n(y),n(x))) = T).  

479 [para_into,287,272,demod,275,3] (i(n(i(x,y)),x) = T).  

491 [back_demod,23,demod,474,3] (i(n(y),n(x)) = i(x,y)).  

496 [para_from,287,6,demod,3,288,3] (i(n(x),y) = i(n(y),x)).  

500,499 [para_into,479,278] (i(i(i(x,y),n(T)),x) = T).  

509 [para_into,491,278] (i(i(x,n(T)),n(y)) = i(y,x)).  

511 [para_into,491,274] (i(x,n(y)) = i(y,n(x))).  

513 [para_into,491,276] (i(n(x),y) = i(i(y,n(T)),x)).  

520 [para_from,491,4] (i(i(n(x),y),i(i(y,n(z)),i(z,x))) = T).  

530 [para_from,491,113,demod,234] (i(x,y) = i(n(y),n(x))).  

533 [para_from,491,49] (i(n(x),i(i(y,x),n(y))) = T).  

536,535 [para_from,491,6] (i(i(n(x),n(y)),n(y)) = i(i(x,y),n(x))).  

540 [para_into,496,276] (i(x,y) = i(n(y),i(x,n(T)))).  

565 [para_from,496,6] (i(i(n(x),y),x) = i(i(x,n(y)),n(y))).  

569 [para_into,511,276] (i(x,y) = i(i(y,n(T)),n(x))).  

598 [para_from,511,6] (i(i(x,n(y)),n(x)) = i(i(n(x),y),y)).  

624,623 [para_into,530,105] (i(n(y),n(i(i(y,x),x))) = i(x,y)).  

627 [para_into,530,6] (i(n(x),n(i(y,x))) = i(i(x,y),y)).  

647 [para_from,530,4] (i(i(x,y),i(i(n(z),n(y)),i(x,z))) = T).  

723 [para_into,289,278] (i(i(x,i(y,n(T))),i(y,n(x))) = T).  

726,725 [para_into,289,530] (i(n(i(x,n(y))),n(i(y,n(x)))) = T).  

771 [para_from,441,6,demod,3,624] (i(i(y,x),n(i(i(x,y),y))) = n(x)).  

808 [para_into,509,276] (i(i(x,n(T)),y) = i(i(y,n(T)),x)).  

1005 [para_from,533,4,demod,3] (i(i(i(i(x,y),n(x)),z),i(n(y),z)) = T).  

1464,1463 [para_into,229,569,demod,258] (i(i(x,n(y)),i(y,i(x,z))) = T).  

1470,1469 [para_into,229,513,demod,258] (i(i(x,y),i(n(y),i(x,z))) = T).  

2855 [para_into,9,530] (i(i(i(n(b),n(a)),i(b,a)),i(b,a)) != T).  

7827,7826 [para_from,725,627,demod,275,726,3] (n(i(y,n(x))) = i(i(x,n(y)),n(T))).  

8405,8404 [para_into,771,723,demod,1464,3,3,7827] (n(i(y,i(x,n(T))))) = i(i(y,n(x)),n(T))).  

8431,8430 [para_into,771,291,demod,1470,3,8405,3] (n(i(x,y)) = i(i(n(y),n(x)),n(T))).  

10857 [para_from,565,113,demod,106] (i(i(n(y),x),x) = i(i(n(x),y),x)).  

10860 [para_from,10857,233,demod,8431,275,8431,275,322] (i(i(n(x),y),x) = i(i(n(y),x),x)).  

10862 [para_into,10860,274,demod,536] (i(i(x,y),n(x)) = i(i(y,x),n(y))).  

10887 [para_into,10862,496,demod,275] (i(i(n(x),y),y) = i(i(x,n(y)),n(x))).  

10936 [para_from,10887,598,demod,275,8431,275,275,8431,275,275,500,3]  

    (i(i(i(x,y),n(x)),i(i(y,x),n(T))) = y).  

11090 [para_into,520,540,demod,258] (i(i(n(x),y),i(i(x,n(z)),i(z,y))) = T).  

11114 [para_into,1005,647,demod,275,3] (i(n(x),i(i(n(y),z),i(i(z,x),y))) = T).  

11184 [para_into,11090,808,demod,8431,275,3,258,258] (i(i(x,y),i(i(z,x),i(z,y))) = T).  

11202 [para_from,11184,377,demod,3] (i(x,i(i(y,i(x,z)),i(y,z))) = T).  

11213,11212 [para_into,11202,11202,demod,3] (i(i(x,i(y,z)),i(y,i(x,z))) = T).  

11285 [para_from,11212,6,demod,3,11213,3] (i(x,i(y,z)) = i(y,i(x,z))).  

11350 [para_from,11285,2855] (i(b,i(i(i(n(b),n(a)),i(b,a)),a)) != T).  

11460 [para_into,11114,10936,demod,275,8431,275,8431,3,222]  

    (i(x,i(i(i(n(x),n(y)),i(x,y)),y)) = T).

```

```

11462 [binary,11460,11350] .

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


----- statistics -----
clauses input          6
clauses given          768
clauses generated      1012625
demod & eval rewrites  2793093
clauses wt,lit,sk delete 249150
tautologies deleted    0
clauses forward subsumed 761142
  (subsumed by sos)    1097
clauses kept            5897
new demodulators        5564
empty clauses           1
clauses back demodulated 3558
clauses back subsumed   18
sos size                1655
Kbytes malloced         6801

----- times (seconds) -----
run time       2249.42          (run time 0 hr, 37 min, 29 sec)
system time    344.04
input time     0.01
  classify time 0.00
  process input 0.01
para_into time 122.66
para_from time 115.25
pre_process time 1927.16
  demod time   1567.96
  weigh cl time 37.51
  for_sub time 56.03
  renumber time 69.31
  keep cl time 9.72
  print_cl time 0.00
  conflict time 1.43
post_process time 65.32
  back demod time 53.96
  back_sub time 10.91
lex_rpo time    76.22
The job finished   Thu Jun 4 18:15:07 1992

```

7 Conclusion

Years of experimentation with theorem-proving systems [4, 9, 3, 5, 6] have enabled us to accumulate a wide variety of variations and parameters to control our basic theorem-proving algorithm. The typical user need know about only a few of them. This contest forced us to consider how we would set them if there were to be a parameterless version of OTTER.

Most of the above settings just represent common sense. The value of `max_mem` did have to be carefully chosen so that we could get proofs of all of these theorems with the same

value.

We thank Ross Overbeek for proposing this exercise, and we hope that others found it as useful as we did.

References

- [1] E. Lusk and W. McCune. Experiments with Roo, a parallel automated deduction system. In B. Fronhöfer and G. Wrightson, editors, *Parallelization in Inference Systems, Lecture Notes in Artificial Intelligence, Vol. 590*, pages 139–162, New York, 1992. Springer-Verlag.
- [2] E. Lusk, W. McCune, and J. Slaney. Roo—a parallel theorem prover. Tech. Memo MCS-TM-149, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Ill., 1991.
- [3] E. Lusk and R. Overbeek. The automated reasoning system ITP. Tech. Report ANL-84/27, Argonne National Laboratory, Argonne, Ill., April 1984.
- [4] J. McCharen, R. Overbeek, and L. Wos. Problems and experiments for and with automated theorem-proving programs. *IEEE Transactions on Computers*, C-25(8):773–782, August 1976.
- [5] W. McCune. OTTER 2.0 Users Guide. Tech. Report ANL-90/9, Argonne National Laboratory, Argonne, Ill., March 1990.
- [6] W. McCune. What’s New in OTTER 2.2. Tech. Memo ANL/MCS-TM-153, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Ill., July 1991.
- [7] W. McCune and L. Wos. The absence and the presence of fixed point combinators. *Theoretical Computer Science*, 87:221–228, 1991.
- [8] W. McCune and L. Wos. Experiments in automated deduction with condensed detachment. In D. Kapur, editor, *Proceedings of the 11th International Conference on Automated Deduction, Lecture Notes in Artificial Intelligence, Vol. 607*, pages 209–223, New York, June 1992. Springer-Verlag.
- [9] B. Smith. Reference manual for the environmental theorem prover: An incarnation of AURA. Tech. Report ANL-88-2, Argonne National Laboratory, Argonne, Ill., March 1988.