Bibliography

Publications that cover material related to the contents of this book are listed below in order of their date of appearance.


A number of journals publish material on automated theorem proving. First and foremost is the *Journal of Automated Reasoning*, published since 1985 by D. Reidel Publishing Company, P. O. Box 17, 3300 AA Dordrecht, Holland, or 190 Old Derby Street, Hingham, Massachusetts 02043. In addi-
tion, *Artificial Intelligence*, published by North-Holland, *Machine Intelligence*, a series of almost a dozen volumes, which publishes every several years, the *Journal of the Association for Computing Machinery*, and the *IEEE Transactions on Computers* have all played leading roles in the publication of papers on automated theorem proving. The Conference on Automated Deduction is held once a year, and its proceedings are on the cutting edge of developments in the field.

Listed below are a number of other publications closely related to the material in this book.


J.D. Lawrence and J.D. Starkey, Experimental tests of resolution-based theorem proving strategies, CS-74-011, Computer Science Department, Washington State University, April 1974.


Appendix A
Answers to Selected Exercises

Chapter 1

1.2. This problem and the next are meant to familiarize you with the TPTP Problem Library. There are 28 categories of theorems. See Documents/OverallSynopsis in the TPTP Problem Library.

1.3. There are 160 geometry theorems in the directory GEO.

Chapter 2

2.1. (a) \( \forall x: \{\text{positiveinteger}(x) \& \text{perfectsquare}(x)\} \Rightarrow \!
\exists y: \{\text{equal}(\text{times}(\text{plus}(y,1),\text{minus}(y,1)),\text{plus}(x,1))\}

(b) \( \forall x: \{\{\text{positiveinteger}(x) \& \neg \text{prime}(x)\} \Rightarrow \!
\exists y: \{\text{prime}(y) \& \text{divides}(y,x) \& \text{lessthan}(y,x)\}\}

(c) \text{on}(c,b) \lor \neg \exists x: \text{on}(c,x) \& \text{above}(x,b) \Rightarrow \text{above}(c,b)

2.2. See HUMBIRD.WFF. 2.3. See MERMAID.WFF.
2.4. See NATNUMB.WFF. 2.5. There are a total of 21 axioms.
2.6. See Q49C2.THM. 2.7. See Q59W3.THM.
2.8. See CANNIBAL.WFF. 2.9. See PUZZLE8.WFF.

Chapter 3

3.1. (a) \neg E(x,y) \lor E(y,x)

(b) \neg A(x) \lor \neg B(x) \lor \neg C(x) \lor D(x)
212 Appendix A: Answers to Selected Exercises

(c) \(-A(x) \lor B(x)\)
\(-A(x) \lor C(x)\)
\(-A(x) \lor D(x)\)

(d) \(-A(x) \lor B(x) \lor C(x) \lor D(x)\)

(e) \(-A(x) \lor D(x)\)
\(-B(x) \lor D(x)\)
\(-C(x) \lor D(x)\)

(f) - (m) use COMPILE

3.2. Run COMPILE on NATNUMB.THM.
3.3. See S25WOS1.THM.
3.4. See Q01D1.THM.

Chapter 4

4.1. (c) \{f(z)/u, f(z)/x\}, Q(f(z), f(z))
(d) \{a/z, f(a)/x, g(y)/u\}, P(a, f(a), f(g(y)))
(e) \{f(a)/x, f(f(a))/y, f(f(f(a)))/z, f(f(f(f(a))))/v, f(f(f(f(f(a)))))/u\},
\ R(f(a), f(f(a)), f(f(f(a))), f(f(f(f(a))))))

4.2. (c) C: (5c, 6b) ~less(a, u) \lor ~less(f(a), u) \lor less(h(u), u)
(d) C: (7b, 8a) P(x, x) \lor ~Q(x, y) \lor Q(f(x), a)
C: (7c, 8b) P((f(u), f(u))) \lor ~P(f(u), h(f(u))) \lor P(u, v)
(e) C: (9a, 10a) Q(y, h(y)) \lor Q(g(u, a), b) \lor P(u) \lor ~Q(b, h(a))

4.3. (b) C: (12cd) P(a, f(a)) \lor P(f(a), f(b)) \lor Q(f(a), a)
(c) C: (13ac) P(a, f(a)) \lor P(f(a), f(a))
C: (13bc) P(a, a) \lor P(a, f(a))

4.4. k*m. Consider C1 = P(a, x) \lor P(b, x) \lor P(c, x) and C2 = ~P(x, a) \lor ~P(x, b) \lor ~P(x, c).

4.5. k*(k - 1)/2. Consider C = P(x, y, z) \lor P(x, y, a) \lor P(x, b, a) \lor P(c, b, a).

4.7. (a) subsumes and s-subsumes, (b) subsumes and s-subsumes,
(c) subsumes, (d) does not subsume,
(e) does not subsume, (f) does not subsume,
(g) does not subsume, (h) subsumes,
(i) does not subsume, (j) does not subsume.

4.9. Consider C1 = ~P(x) \lor P(f(x)) and C2 = ~P(x) \lor P(f(f(x)))
Chapter 5

5.1. Herbrand universe elements: 1 a 2 h(a,a) 3 h(a,h(a,a)) 4 h(h(a,a),a)
5 h(h(a,a),h(a,a)) 6 h(a,h(a,h(a,a))) 7 h(a,h(h(a,a),a)) 8 h(a,h((a,a),h(a,a)))
9 h(h(a,a),h(a,h(a,a))) 10 h(h(a,a),h(h(a,a),a)) 11 h(h(h(a,a),a),h(h(a,a),a))
12 h((a,a),h(a,a)) 13 h(a,h(a,a),h(a,a)) 14 h(h((a,a),h(a,a)),h(a,a,a)))
15 h(h(a,a),h(h(a,a),a),16 h(h(a,h(a,a)),h(h(a,a),h(a,a)))
17 h(h(a,a),h(a,a),h(a,a)) 18 h(h(h(a,a),a),h(a,a)) 19 h(h(h(a,a),a),h(h(a,a),a)))
20 h(h(h(a,a),a),h(h(a,a),a))

Herbrand base elements: 1 P(a,a) 2 Q(a,a) 3 P(a,h(a,a)) 4 Q(a,h(a,a))
5 P(h(a,a),a) 6 Q((a,a),a) 7 P(a,h(a,h(a,a))) 8 Q(a,h(a,h(a,a))) 9 P(h(a,a),h(a,a))
10 Q(h(a,a),h(a,a)) 11 P(h(a,h(a,a)),a) 12 Q(h(a,h(a,a)),a) 13 P(a,h(h(a,a),a))
14 Q(a,h(h(a,a),a)) 15 P(h(a,a),h(h(a,a),a)) 16 Q(h(a,a),h(h(a,a),a)))
17 P(h(a,a),h(h(a,a),a)) 18 Q(h(a,a),h(h(a,a),a)) 19 P(h(h(a,a),a),a)
20 P(h(h(a,a),a),a)

5.2. There are two terms in the Herbrand universe: HU = {a,e}. There are
31 atoms in the Herbrand base.

5.4.

1: (1/1) (2/2) (3/3) 3: (1/1) (2/2) (3/3) 5: (1/1) (2/2) (3/3)
2: (1/1) (2/2) (3/3) 4: (1/1) (2/2) (3/3) 6: (1/1) (2/2) (3/3)

5.5. Use atom P(a).

5.6. Use atoms P(a), P(g(a)), P(g(g(a))), and P(g(g(g(a))))

5.7. Use atoms Equal(e,a), P(e,a,a), and P(e,a,e).
Chapter 6

6.1. $\forall x: \neg \text{Prime}(x) \Rightarrow \{y: \text{Divides}(y,x) \& \text{Prime}(y) \& \text{Less}(y,x)\}$

6.2. Step 1: Form resolvent $C8$: $(1c, 2c) \neg P \lor \neg Q$. This yields the semantic tree:

Step 2: Form resolvent $C9$: $(3c, 4c) \neg P \lor Q$. This gives the semantic tree:
Step 3: Form resolvent $C_{10}$: $(5c,6c) \ P \ I \ -Q$. This gives the semantic tree:

Steps 4, 5, and 6: (The semantic trees are omitted for these last three steps.)
Form resolvents
$C_{11}$: $(8b,9b) \ -P$
$C_{12}$: $(10b,7b) \ P$
$C_{13}$: $(11a,12a) \ \emptyset$

6.3.

6.4.

6.5. This is a difficult problem, but lots of fun!

6.6. $2^{D+1} - 2 - D$
Chapter 7

7.2. Examine the source code in atom.c.

7.3. Run HERBY on STARK036.THM using the d2 option. Whenever the execution of the program halts, enter a carriage return. From the printout, you can then construct the tree by hand (36 nodes in the tree) and show where the tree-pruning heuristic of Section 7.2.5 comes into play.

Chapter 8

8.1. The heuristic was used.

8.4. Examine the proof found by THEO and see if that gives you any ideas for selecting atoms.

Chapter 9

9.1. Check your results by running THEO with the options p20 k0 h0 n0 m0 z0. The results can be found in the file B.T0. (Note: if your computer is slow, the results may be placed in B.T1, where the 1 indicates your computer took one second to prove the theorem; a 0 indicates that less than one second was
necessary.) Two iterations are necessary, and seven nodes are generated.

9.2. Check your results by running THEO with the options p20 k0 h0 n0 z1. THEO carries out three iterations and generates seven clauses when proving B.THM. THEO carries out five iterations and generates 18 clauses when proving C.THM.

9.3. Check your results by running THEO with the options p20 k0 h0 z1 b0. THEO carries out one iterations and generates three clauses when proving B.THM. THEO carries out three iterations and generates 11 clauses when proving C.THM.

9.4. \( P(x,f(a)) \Rightarrow 15, \neg Q(y,f(x)) \Rightarrow -19 \quad \neg P(x,y) \Rightarrow -103, \neg Q(z,y) \Rightarrow -106, Q(a,x) \).

9.5. For \( \text{LITmaxbase} = 2 \), the following is a counterexample: 1 \( p \lor s \), 2 \( \neg s \lor r \), 3 \( \neg p \lor r \), 4 \( \neg p \lor t \), 5 \( \neg t \lor \neg r \), 6 \( \neg p \lor \neg r \). Now, try \( \text{LITmaxbase} = 3 \) for yourself!

9.6. There are four constants, \( e,a,b,c \). Clauses 1, 2, 6, 16, and 17 contain one variable; for each there are five hash codes entered in the clause_hash_table. For clauses 3 which has two variables, there are 25 hash codes entered, and for clauses 20, 21, 22, and 23 there is one each.

Chapter 10

10.1. (b) Sixteen entries are made in clause_hash_table. (c) The two unit base clauses, UNAAB and EAB, account for two, then during the search, entries were made for clauses \( \neg EAB \) (1 entry), Sxx (4 entries corresponding to Sxx, Syy, SAA, and SBB), Exx (4 entries), SBA (1 entry), \( \neg SAB \) (1 entry), MFABA (coming from the clause SAx | MFxA after literal SAx was resolved away as discussed in Section 9.14), MFABB (1 entry), and SAB (1 entry) for a total of 16. (Note: this problem is impossible to answer completely without understanding some of the source canse in the file search.c.)

10.7. The four theorems are in the directory THEOREMS. If you create PROVEALL1 as shown here, all the theorems can be attempted without user interaction. An upper limit of 1220 seconds is placed on each attempt to find a proof.

STARK075.THM z0 h0 n0 t1200
STARK075.THM z0 h0 n1 t1200
STARK075.THM z0 h1 n0 t1200
10.8. The theorem can be found in the directory `THMSMISC`.

10.9. First, convert the wffs `cannibal.wff` and `puzzle8.wff` in the directory `THMSMISC` to clauses and then use `THEO` to obtain proofs.

10.10, 10.11. The theorems can be found in the directory `THMSMISC`.

**Chapter 13**

13.1. These are clauses that are not equality substitution clauses.

13.2. The first two literals of clause `88350` are factored and then the result resolved with clause `88349`.

13.3. The clauses \( \neg \text{Equal}(x,y) \lor \text{Equal}(y,x) \) and \( \neg \text{Equal}(x,y) \lor \neg \text{p2}(u,v,x) \lor \text{p2}(u,v,y) \).
Appendix B
List of Wffs and Theorems in the Directories
WFFS, THEOREMS, GEOMETRY, and THMSMISC

The directory WFFS contains 21 files, each containing at least one wff:

1. CANNIBAL.WFF ; Exercises 2.8, 10.9: missionaries and cannibals
2. EXCOMP1.WFF ; Chapter 3.1: running example
3. EXCOMP2.WFF ; Example 1, Section 3.3
4. EXCOMP3.WFF ; Example 2, Section 3.3
5. EXCOMP4.WFF ; Example 3, Section 3.3
6. EXCOMP5.WFF ; Example 4, Section 3.3
7. EXER3F.WFF ; Exercise 3.1f
8. EXER3G.WFF ; Exercise 3.1g
9. EXER3H.WFF ; Exercise 3.1h
10. EXER3I.WFF ; Exercise 3.1i
11. EXER3J.WFF ; Exercise 3.1j
12. EXER3K.WFF ; Exercise 3.1k
13. EXER3L.WFF ; Exercise 3.1l
14. EXER3M.WFF ; Exercise 3.1m
15. EXER3N.WFF ; Exercise 3.1n
16. EXER3O.WFF ; Exercise 3.1o
17. HUMBIRD.WFF ; Exercise 2.2
18. MERMAID.WFF ; Exercise 2.3
19. NATNUMB.WFF ; Exercise 2.4, 3.2
20. PUZZLE8.WFF ; Exercises 2.9, 10.9: the 8-puzzle
21. WOLVES.WFF ; Exercise 10.12

The directory THEOREMS contains 85 files, 84 each containing one theorem of the Stickel test set. One file is intended for use in proving the Stickel set without user intervention. These theorems have been used by several researchers when testing their theorem-proving programs.
220

Appendix B: List of Wffs and Theorems

1. S01BURST.THM
2. S02SHORT.THM
3. S03PRIM1.THM
4. S04HASP1.THM
5. S05HASP2.THM
6. S06ANCES.THM
7. S07NUM1.THM
8. S08GRP1.THM
9. S09GRP2.THM
10. S10EW1.THM
11. S11EW2.THM
12. S12EW3.THM
13. S13ROB1.THM
14. S14ROB2.THM
15. S15MICHI.THM
16. S16QW.THM
17. S17MQW.THM
18. S18DBABH.THM
19. S19APABH.THM
20. S20FLEI1.THM
21. S21FLEI2.THM
22. S22FLEI3.THM
23. S23FLEI4.THM
24. S24FLEI5.THM
25. S25WOS1.THM
26. S26WOS2.THM
27. S27WOS3.THM
28. S28WOS4.THM
29. S29WOS5.THM
30. S30WOS6.THM
31. S31WOS7.THM
32. S32WOS8.THM
33. S33WOS9.THM
34. S34WOS10.THM
35. S35WOS11.THM
36. S36WOS12.THM
37. S37WOS13.THM
38. S38WOS14.THM
39. S39WOS15.THM
40. S40WOS16.THM
41. S41WOS17.THM
42. S42WOS18.THM
43. S43WOS19.THM
44. S44WOS20.THM
45. S45WOS21.THM
46. S46WOS22.THM
47. S47WOS23.THM
48. S48WOS24.THM
49. S49WOS25.THM
50. S50WOS26.THM
51. S51WOS27.THM
52. S52WOS28.THM
53. S53WOS29.THM
54. S54WOS30.THM
55. S55WOS31.THM
56. S56WOS32.THM
57. S57WOS33.THM
58. STARK005.THM
59. STARK017.THM
60. STARK023.THM
61. STARK026.THM
62. STARK028.THM
63. STARK029.THM
64. STARK035.THM
65. STARK036.THM
66. STARK037.THM
67. STARK041.THM
68. STARK055.THM
69. STARK065.THM
70. STARK068.THM
71. STARK075.THM
72. STARK076.THM
73. STARK087.THM
74. STARK100.THM
75. STARK103.THM
76. STARK105.THM
77. STARK106.THM

58. STARK005.THM ; Lawrence and Starkey #5, Nilsson (1970), p. 220.
59. STARK017.THM ; There exist infinitely many primes.
60. STARK023.THM ; In a group with right inverses and right iden-
tity, every element has a left inverse.
61. STARK026.THM ;
62. STARK028.THM ; (a + b) + c = a + (b + c)
63. STARK029.THM ; (a - b) + c = (a + c) - b
64. STARK035.THM ;
65. STARK036.THM ; In a group, (x times y) inverse equals x inverse
times y inverse.
66. STARK037.THM ; In a ring, x*0 = 0.
67. STARK041.THM ; Equality is symmetric.
68. STARK055.THM ; Kleene, Intro. to Metamathematics.
69. STARK065.THM ; If x is less than y and y is less than or equal to
z, then x is less than z.
70. STARK068.THM ; 0 < x'.
71. STARK075.THM ;
72. STARK076.THM ; If x < y, then y < x.
73. STARK087.THM ; If z ≠ 0, then if zx < yz, x < y.
74. STARK100.THM ; If x = y, then every member of x is also a
member of y.
75. STARK103.THM ; Union is idempotent.
76. STARK105.THM ; If (x union y) = y, then x is a subset of y.
77. STARK106.THM ; x is the union of x and y.
Appendix B: List of Wffs and Theorems

78. STARK108.THM ; Intersection is associative.
79. STARK111.THM ; If the intersection of x and y is x, then x is a subset of y.
80. STARK112.THM ; x union (y intersection z) equals (x intersection y) union (x intersection z).
81. STARK115.THM ; Intersection of x & (y - x) is empty.
82. STARK116.THM ; If x is in y, then z - y is in z - x.
83. STARK118.THM ; (z - x) union (z - y) = z - (x intersection y).
84. STARK121.THM ; x intersection y equals x - (x - y).

The directory GEOMETRY contains 66 files, each containing one theorem. These theorems on Euclidean geometry are the subject of Art Quaife’s paper, “Automated development of Tarski’s geometry” (listed in the bibliography). These theorems are based on the axioms presented in Section 2.6. Each successive theorem uses all the previous ones as axioms.

1. Q01D1.THM ; ordinary reflexivity of equidistance
2. Q02D2.THM ; equidistance is symmetric between argument pairs
3. Q03D3.THM ; equidistance is symmetric with argument pairs
4. Q04D4.THM ; corollaries to Q02D2.THM and Q03D3.THM
5. Q05D5.THM ; ordinary transitivity of equidistance
6. Q06E1.THM ; NULL extension
7. Q07B0.THM ; corollary to axiom A4.1
8. Q08R2.THM ; corollaries to axioms A4
9. Q09R3.THM ; corollaries to Q06E1.THM
10. Q10R4.THM ; u is the only fixed point of R(u,v)
11. Q11D7.THM ; all NULL segments are congruent
12. Q12D8.THM ; addition of equal segments
13. Q13D9.THM ; unique extension
14. Q14D10A.THM ; corollaries to Q13D9.THM
15. Q14D10B.THM ;
16. Q14D10C.THM ;
17. Q15R5.THM ; congruence for double reflection
18. Q16R6.THM ; R is an involution
19. Q17T3.THM ; v is between u and v
20. Q18B1.THM ; corollary with axiom A6
21. Q19T1.THM ; between is symmetric in its outer arguments
22. Q20T2.THM ; u is between u and v
23. Q21B2.THM ; antisymmetry of between in first two arguments
24. Q22B3.THM ; corollary to Q21B2.THM
Appendix B: List of Wffs and Theorems

25. Q23T6.THM ; given three distinct points u, v, w, if v is
    ; between u and w, then u is not between v and
    ; w, and w is not between u and v.
26. Q24B4.THM ; first inner transitivity property of betweenness
27. Q25B5.THM ; corollary using symmetry
28. Q26B6.THM ; first outer transitivity property of betweenness
29. Q27B7.THM ; second outer transitivity property of betweenness
30. Q28B8.THM ; second inner transitivity property of betweenness
31. Q29B9.THM ; corollary using symmetry
32. Q30E2.THM ; there are at least three distinct points
33. Q31E3.THM ; a segment can be lengthened
34. Q32B10.THM ; inner points of a triangle
35. Q33D11.THM ; corollary to outer five-segment axiom
36. Q34D12.THM ; second inner five-segment theorem
37. Q35D13.THM ; subtraction of line segments
38. Q36D14.THM ; first inner five-segment theorem
39. Q37D15.THM ; corollary
40. Q38I2A.THM ; theorem of point insertion
41. Q38I2B.THM ; " "
42. Q38I2C.THM ; " "
43. Q39I3.THM ; insertion identity
44. Q40I4.THM ; insert respects congruence in its last two arguments
45. Q41B11.THM ; theorem of similar situations
46. Q42B12.THM ; first outer connectivity property of betweenness
47. Q43B13.THM ; second outer connectivity property of betweenness
48. Q44T7.THM ; given two distinct points w, x, and two distinct
    ; points u, v on the segment wx, to the left of w,
    ; then either v is between u and w, or u is
    ; between v and w.
49. Q45T9.THM ; first inner connectivity property of betweenness
50. Q46B14.THM ; second inner connectivity property of betweenness
51. Q47T8.THM ; five-point theorem
52. Q48B15.THM ; unique endpoint
53. Q49C2.THM ; collinearity: corollary using symmetry
54. Q50T10.THM ; collinearity is invariant under permutation of
    ; arguments
55. Q51T11.THM ; the three points p', p", and p"" are not collinear
56. Q52C3.THM ; any two points are collinear
57. Q53C4.THM ; theorem of similar situation for C(u,v,w)
58. Q54T12.THM ; given three distinct collinear points u,v,w, and
    ; a fourth point x collinear with u and v, then x is
    ; collinear with uw and collinear with vw.
Appendix B: List of Wffs and Theorems

59. Q55C5.THM ; additional versions using symmetry
60. Q56T13.THM ; given two distinct points \( u, v \) and three points
   \( w, w', w'' \) which are collinear with \( uv \), then
   \( w, w', w'' \) are collinear.
61. Q57W1A.THM ; lemma: under the hypothesis of the bisecting
   diagonal theorem, the points \( u, v, w \) cannot be
   collinear
62. Q57W1B.THM ;
63. Q57W1C.THM ;
64. Q58W2A.THM ; corollaries
65. Q58W2B.THM ;
66. Q59W3.THM ; the diagonals of a nondegenerate rectangle
   bisect each other

The directory THMSMISC contains 32 files, each containing one theorem. These are miscellaneous theorems that appear in the text or in various research papers.

1. CHANG1.THM 17. LOVE39A.THM
2. CHANG2.THM 18. LOVE39B.THM
3. CHANG3.THM 19. LOVE39C.THM
4. CHANG4.THM 20. LIFSCHTZ.THM
5. CHANG5.THM 21. EQUAL1.THM
6. CHANG6.THM 22. EQUAL2.THM
7. CHANG7.THM 23. A.THM
8. CHANG8.THM 24. B.THM
9. CHANG9.THM 25. C.THM
10. AGATHA.THM 26. FACT.THM
11. LION.THM 27. SQROOT.THM
12. KNGTS.THM 28. HU1.THM
13. WOLVES1.THM 29. HU2.THM
14. WOLVES2.THM 30. LINEAR.THM
15. WOLVES3.THM 31. HASH.THM
16. SAM.THM 32. EX2.THM
Index

=> (implication), 7, 9
@ (universal quantifier), 8
! (existential quantifier), 8
\forall (universal quantifier), 8
\exists (existential quantifier), 8
& (logical and), 7, 9, 24
( (left parenthesis), 8, 101
) (right parenthesis), 8, 101
* (asterisk), 146
, (comma), 8
. (period), 8, 24, 146
: (colon), 8
; (semicolon), 24, 97, 139
<=> (if and only if), 7, 9
> (greater than), 148
? (question mark), 100, 143
[ (left bracket), 8
\mid (logical or), 7, 101
\} (right bracket), 8
~ (negation), 7, 9, 21
\oplus (exclusive-or), 123, 124

A.THM, 55, 64, 115–120
AGATHA.WFF, 24–25, 156–157
AI (see artificial intelligence)
Algorithm 5.1 Construction of a closed semantic tree, 48–50
Algorithm 6.1 Obtaining a resolution-refutation proof, 59–63
Algorithm 6.2 Obtaining a linear proof, 64–70
Algorithm 7.1 Semantic tree construction, 94
Almulla, Mohammed, 50
Artificial intelligence, 18
Atom, 44
Aunt Agatha riddle, 156–157
Axioms, equality, 13, 16, 145, 149
Axioms, equality substitution, 13, 16–17, 18, 101, 108, 145, 149
Axioms, natural numbers, 18

B.THM, 137
Binary factoring, 30, 37–38, 40
Binary resolution, 29–30, 33–38, 40
resolve away, 35
resolve on, 35
resolve together, 35
Bindings, 31, 32
Bound, 31, 32

C programming language, 1
C.THM, 137
CADE (see Conference on Automated Deduction)
CD-ROM, 5
Clause, 23
ancestor, 64
base, 40
conjunctive normal form, 23, 28
disjunction, 33
ground, 31, 64, 79–80
inferred, 40
Index

GE0041-2, 182, 185–188, 195–199, 204, 205
GE0059-2, 182, 188–190, 199–204, 205

GEOMETRY, 5
Group theory, 12
axioms, 12–13

HASH.THM, 123–124, 137
Herbrand base, 44, 45
canonical order, 44, 47, 48
interpretation on, 45–47
Herbrand universe, 43–44
canonical order, 43–44

HERBY, 1, 2, 3–4, 12, 13, 21, 29, 35, 36, 39, 43,
85–95, 97–111, 161–172, 179, 181
atom selection heuristic, 85–91, 92
atom selection heuristic 0 (ASH0), 91, 92, 94
atom selection heuristic 1 (ASH1), 85, 92, 94
atom selection heuristic 2 (ASH2), 86, 92, 94, 105
atom selection heuristic 3 (ASH3), 87, 92, 94
atom selection heuristic 4 (ASH4), 89, 92, 93, 94, 103, 171
atom selection heuristic 5 (ASH5), 89–90, 92, 94
atom selection heuristic 5a (ASH5a), 89–90, 92, 102
atom selection heuristic 5b (ASH5b), 89, 90, 92
atom selection heuristic 5c (ASH5c), 89, 90, 92
base clause resolution heuristic, 93
bounding the number of literals, 93, 99
bounding the number of terms, 93, 99, 100
clear_var_substitution_table, 172
constant selection heuristic, 91
creating executable version, 3
extended base set, 102
function linkage, 162–164
generate_clause, 172
hash codes for clauses, 94
hash codes for literals, 94
heuristics, 85–94, 95
greedy, 90
hyperresolvable quad, 1–1–1–3, 87
hyperresolvable trio, 1–1–2, 86–87
input file, 97
input file, comments, 97
main functions, 164–165
add_clause, 165
atom_sel_heuristics, 164, 165
build_tree, 165
calculate_hash_codes, 164
clear_failure_pair_info, 164
CompileTptp, 164
compile_clauses, 165
enter_theorem_name, 164
init_rand_hash_array, 164
init_search_space, 164
main, 161, 164
prepare_clauses, 164
resolution_stage, 164–165
supervise_build_tree, 165

major arrays, 170–171
branch, 171
clist, 85, 86, 87, 89, 90, 91, 92, 93, 100, 102, 103, 171
cspace, 170, 171
depth, 171
fail, 171
fail_count, 171
hb_list, 170, 171
hb_space, 170, 171
rand_array, 171
unit_list, 85, 89, 90, 92, 93, 102, 103, 106
unit_space, 170, 171
makefile, 3, 5, 164

obtaining a resolution–refutation proof, 94, 99, 107–111
obtaining help, 100
occurs, 172
ordering clauses, 91
on clist, 91
on unit_list, 93
output file, 98
.h suffix, 98
.Hx suffix, 98
naming, 98

Phase 0 (see base clause resolution heuristic),
Phase 1 (construct semantic tree), 93, 94
pop_stack, 172
prepare_clauses, 99
printout, 100–111
printout with r1 option, 107–111
resolvable pair, 1–1, 85–86
running (see HERBY, using)
semantic tree pruning heuristic, 94, 95
<table>
<thead>
<tr>
<th>Term</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>source code, 161–172</td>
<td></td>
</tr>
<tr>
<td>source files, 3, 5, 161–164</td>
<td></td>
</tr>
<tr>
<td>affc.c, 161, 164</td>
<td></td>
</tr>
<tr>
<td>atom.c, 161</td>
<td></td>
</tr>
<tr>
<td>build.c, 161</td>
<td></td>
</tr>
<tr>
<td>clause.h, 161</td>
<td></td>
</tr>
<tr>
<td>const.h, 161</td>
<td></td>
</tr>
<tr>
<td>equal.c, 161, 164</td>
<td></td>
</tr>
<tr>
<td>hdump.c, 161</td>
<td></td>
</tr>
<tr>
<td>herby.c, 99, 161, 164</td>
<td></td>
</tr>
<tr>
<td>hgen.c, 161</td>
<td></td>
</tr>
<tr>
<td>hhash.c, 161</td>
<td></td>
</tr>
<tr>
<td>infer.c, 161</td>
<td></td>
</tr>
<tr>
<td>tptpi.c, 161, 164</td>
<td></td>
</tr>
<tr>
<td>unify.c, 161, 164</td>
<td></td>
</tr>
<tr>
<td>standardize, 172</td>
<td></td>
</tr>
<tr>
<td>time consumed, 98</td>
<td></td>
</tr>
<tr>
<td>time limit, 99</td>
<td></td>
</tr>
<tr>
<td>unify, 172</td>
<td></td>
</tr>
<tr>
<td>unit-generating pair, 88</td>
<td></td>
</tr>
<tr>
<td>unit-generating trio, 88</td>
<td></td>
</tr>
<tr>
<td>user interaction, 100</td>
<td></td>
</tr>
<tr>
<td>user options, 98–100</td>
<td></td>
</tr>
<tr>
<td>default, 3, 145</td>
<td></td>
</tr>
<tr>
<td>d2, 91, 100–107</td>
<td></td>
</tr>
<tr>
<td>examples, 100</td>
<td></td>
</tr>
<tr>
<td>i1, 91, 92, 99, 111</td>
<td></td>
</tr>
<tr>
<td>observing construction, 99</td>
<td></td>
</tr>
<tr>
<td>proving a set of theorems, 99</td>
<td></td>
</tr>
<tr>
<td>using, 3–4, 97–111</td>
<td></td>
</tr>
<tr>
<td>HERBY/THEO theorem format, 97, 139</td>
<td></td>
</tr>
<tr>
<td>HERBYSYC, 3, 5, 161</td>
<td></td>
</tr>
<tr>
<td>HU1.THM, 43, 44, 45–47, 60–61</td>
<td></td>
</tr>
<tr>
<td>HU2.THM, 43, 44, 52, 61–63</td>
<td></td>
</tr>
<tr>
<td>HU3.THM, 44, 45–47, 48–50</td>
<td></td>
</tr>
<tr>
<td>HU4.THM, 51</td>
<td></td>
</tr>
<tr>
<td>Inference, 12</td>
<td></td>
</tr>
<tr>
<td>Inference procedure, 12, 29–42</td>
<td></td>
</tr>
<tr>
<td>Instance, 31</td>
<td></td>
</tr>
<tr>
<td>common, 31</td>
<td></td>
</tr>
<tr>
<td>ground, 31, 64, 79, 100, 131</td>
<td></td>
</tr>
<tr>
<td>most common, 33</td>
<td></td>
</tr>
<tr>
<td>substitution, 31, 33</td>
<td></td>
</tr>
<tr>
<td>Interpretation, 12, 40</td>
<td></td>
</tr>
<tr>
<td>fails (see fails to satisfy)</td>
<td></td>
</tr>
<tr>
<td>fails to satisfy, 45</td>
<td></td>
</tr>
<tr>
<td>partial, 47, 48</td>
<td></td>
</tr>
<tr>
<td>satisfies, 45</td>
<td></td>
</tr>
<tr>
<td>Iteration, 114, 115, 122, 146</td>
<td></td>
</tr>
<tr>
<td>Knights and knaves riddle, 157, 158</td>
<td></td>
</tr>
<tr>
<td>KNTS.THM, 157, 158</td>
<td></td>
</tr>
<tr>
<td>Lemma 6.1 On linear merge-free derivations, 71–74</td>
<td></td>
</tr>
<tr>
<td>Lemma 6.2 A clause form the negated conclusion must appear, 79</td>
<td></td>
</tr>
<tr>
<td>Lemma 6.3 The lifting lemma, 79–80</td>
<td></td>
</tr>
<tr>
<td>Lifting lemma, 79–80</td>
<td></td>
</tr>
<tr>
<td>Linear proof, 68–72, 83, 114–119, 135, 136, 137 restrictions on form, 71–79</td>
<td></td>
</tr>
<tr>
<td>Linear-merge proof, 74–78, 83, 116–118, 135, 136, 141</td>
<td></td>
</tr>
<tr>
<td>Linear-merge-nc proof, 79, 119, 120, 137</td>
<td></td>
</tr>
<tr>
<td>Linear-neg proof, 79, 118–119, 135, 136, 141, 142, 145</td>
<td></td>
</tr>
<tr>
<td>LINEAR.THM, 67–70</td>
<td></td>
</tr>
<tr>
<td>LINEAR_MERGE.THM, 75–79</td>
<td></td>
</tr>
<tr>
<td>LINUX, 1</td>
<td></td>
</tr>
<tr>
<td>Lion and unicorn problem, 157, 159</td>
<td></td>
</tr>
<tr>
<td>LION.THM, 157, 159</td>
<td></td>
</tr>
<tr>
<td>Literal, 8, 9</td>
<td></td>
</tr>
<tr>
<td>index, 92</td>
<td></td>
</tr>
<tr>
<td>complementary, 124</td>
<td></td>
</tr>
<tr>
<td>disjunction, 23, 33</td>
<td></td>
</tr>
<tr>
<td>resolved on, 35, 72, 73, 74, 77</td>
<td></td>
</tr>
<tr>
<td>Logical operators, 7, 9, 21</td>
<td></td>
</tr>
<tr>
<td>Logically follow, 10–12, 40</td>
<td></td>
</tr>
<tr>
<td>MAXVARS, 123</td>
<td></td>
</tr>
<tr>
<td>Mgu, (see Unifier, most general)</td>
<td></td>
</tr>
<tr>
<td>Missionaries and cannibals problem, 18, 157</td>
<td></td>
</tr>
<tr>
<td>Modus ponens, 38</td>
<td></td>
</tr>
<tr>
<td>negated conclusion, 25, 97, 139</td>
<td></td>
</tr>
<tr>
<td>O (NULL clause), 30</td>
<td></td>
</tr>
<tr>
<td>OCTOPUS, 181</td>
<td></td>
</tr>
<tr>
<td>Otter, 181, 190–204</td>
<td></td>
</tr>
<tr>
<td>o_maxlit, 121, 134</td>
<td></td>
</tr>
<tr>
<td>o_maxvar, 121</td>
<td></td>
</tr>
<tr>
<td>Pelletier, Jeff, 156, 158</td>
<td></td>
</tr>
<tr>
<td>Perl, 6</td>
<td></td>
</tr>
<tr>
<td>Predicate, 6, 8, 30, 32, 33, 44, 122, 123</td>
<td></td>
</tr>
<tr>
<td>Predicate calculus (first-order), 1, 7, 64, 79</td>
<td></td>
</tr>
<tr>
<td>Proof by contradiction, 12, 42</td>
<td></td>
</tr>
<tr>
<td>PROVEALL, 99, 143, 156</td>
<td></td>
</tr>
<tr>
<td>Punctuation marks, 7, 8</td>
<td></td>
</tr>
<tr>
<td>Q07BO.THM, 54, 64, 79–80, 82</td>
<td></td>
</tr>
</tbody>
</table>
Index

Quaife, Art, 5, 14
Quantifier, 7, 8, 22
   existential, 8, 9
   universal, 8, 9

Resolution-refutation proof, 53–83
   graph, 54–57, 64
      branches in, 54
      nodes in, 54
   tree, 64
      principal path, 64, 66, 67
   depth, 58
   length, 58
   main line, 125, 146, 149
   search procedure, 53
Resolution-refutation search tree, 114
   ancestor node, 116, 120
   extended search region, 120, 133
   node, 114
   nonroot, 114
   root, 114
   search horizon, 120, 132, 133, 135, 136
Resolution-refutation search, depth-first search, iteratively deeping, 114–116
   repeating, 125–127
Resolvent, (see binary resolvent)

S, 148
s-subsumption, (see simple subsumption)
S27WOS3A.THM, 52
S31WOS7.THM, 123, 125–127
S39WOS15.THM, 97–98, 128–129, 137, 139–140, 156
S44WOS20.THM, 111, 145–148
S46WOS22.THM, 156
S50WOS26.THM, 149–155
Schubert, Len, 156, 158
Scope, 9
Search, breadth-first, 115
   branch, 47, 48, 51, 102
   canonical, 48
   closed, 47–51, 58–63
   constructing a closed, 48–50
   failure node, 48, 103, 107
   node, 47
   noncanonical, 48, 50–51
   nonterminal node, 47
   root node, 47
   terminal node, 47, 50
Software, accompanying, 5

Sound (logically), 12, 40, 41
Source of resolved-on literal, 72, 73, 77
SORO075.THM, 83–84, 156
Standardize apart, 35, 37
STARK017.THM, 28, 55–57, 64, 82, 155–156
STARK086.THM, 144
STARK075.THM, 156
STARK087.THM, 155
STARK103.THM, 100, 101–107, 107–111, 155, 156
STARK108.THM, 111
STARK118.THM, 111, 156
State-space search problems, 18
Steamroller problem, 158–159
Stickel test set, 5, 50
Substitution, 31, 32, 33, 39
   composition of, 32, 34
   distinct, 32
   NULL, 31, 39
Subsume, 32, 33, 39
Subsumption, simple, 39, 41
Sutcliffe, Geoff, 181
Suttner, Christian, 181
Syllogism, 17
Tarski, Alfred, 14
Tautology, 10, 11
Term, 8, 31, 32, 43, 44, 93, 99, 122, 167
TGTP (see The Great Theorem Prover)
The Great Theorem Prover, 181
add_clause_to_ht, 155
assembling the proof, 133–134, 135
bounding the number of literals, 121, 135, 141, 142, 145
bounding the number of terms, 121, 135, 141, 142, 145
bounding the number of variables, 121, 135, 141, 142, 145
clause hash code, 122, 124, 141, 142
clause_hash_table, 122, 135
constants in extended search region, 133, 135
creating executable version, 4
eliminating tautologies, 133, 135
extended search strategy, 120, 135, 136, 141, 142, 145
extending search on merge clauses, 132, 135
extending search on clauses from negated conclusion, 132, 135
elim.c, 173
equal.c, 173
gen.c, 173
hash.c, 173, 179
inter.c, 173
pass2.c, 173, 180
search.c, 155, 173
theo.c, 173
tptpi.c, 173, 174
unify.c, 173
special consideration of Equal predicate, 133
time consumed, 98, 148
time limit, 142, 145
unify, 180
unit clauses, 125
unit hash table resolution, 128–129, 130, 131, 149
user interaction during search, 144, 145, 156
user options, 140–143
default, 4
examples, 144
information displayed, 140, 142–143
printout with d1, 149–155
proving a set of theorems, 140, 143
user options, search strategy, 140, 141–142
using, 139–160
using simple subsumption, 132, 135, 141
verify proof, 156
Theorem, 1–2, 11–12
axioms, 1, 2, 11, 12, 40, 91, 139
conclusion, 1, 2, 8, 11, 12
givens, 11
hypothesis, 1, 8, 11, 12, 40
if–then, 1
negated conclusion, 12, 79, 91, 132
to prove, 1, 11
Theorem 5.1 Herbrand’s Theorem (Version 1), 47
Theorem 5.2 Herbrand’s Theorem (Version 2), 48
Theorem 6.1 Every theorem has a resolution–refutation proof, 59
Theorem 6.2 Every theorem has a linear proof, 64
Theorem 6.3 Every theorem has a linear-merge proof, 74–79
Theorem 6.4 Every theorem has a linear-nc proof, 79
Theorem 6.5 Every theorem has a linear-merge-nc proof, 79

THEOREMS, 5, 155
THEOSC, 4, 5, 173
THMSMISC, 5
Thousands of Problems for Theorem Provers (see TPTP Problem Library)
TP, 5
TPTP Problem Library, 6, 100, 181
GEO directory, 6, 100
include files, 97, 139
obtaining, 6
TPTP Problem Library format, 97, 99, 100, 139, 142, 161, 164, 173
Truth table, 45–47

Unifier, most general, 33, 34
Unify, 31–32, 33, 34
UNIX, 1, 181

Variable, 8, 9
existentially quantified, 9, 22–23
quantified, 9, 22, 23
universally quantified, 9, 23, 35
normal order, 35, 130, 131
Vine-form proof, 83

Well-formed formula, 7–19
creating, 9
interpretation of, 7, 9–12
fails to satisfy, 10
satisfies, 10, 12, 40
negating, 24, 56
operator precedence, 9
satisfiable, 10, 24
syntax, 7–9
transforming to clause, 21–28
unsatisfiable, 12
wff (see Well-formed formula)
WFFS, 5, 25
WOLVES.WFF, 158–159
www.jessen.informatik.tu-muenchen.de/~tptp/CASC-13/, 181