Programming Report for Resolution Prover

1. Representation of Harmonia theorem in Lisp:

With Answer(x):

```
(defvar *Harmonia* '((1 ((Grandparent x y)) ((Parent x z) (Parent z y)))
                     (2 ((Parent x y)) ((Mother x y)))
                     (3 ((Parent x y)) ((Father x y)))
                     (4 ((Father (Zeus) (Ares))) nil)
                     (5 ((Mother (Hera) (Ares))) nil)
                     (6 ((Father (Ares) (Harmonia))) nil)
                     (7 nil ((Grandparent x (Harmonia)))))) ; Theorem Clause begins here
```

With Answer(X):

```
(defvar *HarmoniaAnswer* '((1 ((Grandparent x y)) ((Parent x z) (Parent z y)))
                     (2 ((Parent x y)) ((Mother x y)))
                     (3 ((Parent x y)) ((Father x y)))
                     (4 ((Father (Zeus) (Ares))) nil)
                     (5 ((Mother (Hera) (Ares))) nil)
                     (6 ((Father (Ares) (Harmonia))) nil)
                     (7 ((Answer x)) ((Grandparent x (Harmonia)))))) ; Theorem Clause begins here
```

2.

	Two-pointer	Unit-preference
Howling Hound	46	27
Drug dealers and customs official	46	21
Coyote and roadrunner	293	84
Harmonia	21	10
My own theorem	6	5

My theorem:

Without Answer(x)

```
(defvar *MyTheorem* '((1 ((VicePresident x)) ((OnTicket x y) (President y)))
                      (2 ((President y)) ((WinElection y)))
                      (3 ((WinElection (Trump))) nil)
                      (4 ((Onticket (Pence) (Trump))) nil)
                      (5    nil ((VicePresident x))))) ; Theorem Clause begins here
```

With Answer(x)

```
(defvar *MyTheo* '((1 ((VicePresident x)) ((OnTicket x y) (President y)))
                      (2 ((President y)) ((WinElection y)))
                      (3 ((WinElection (Trump))) nil)
                      (4 ((Onticket (Pence) (Trump))) nil)
                      (5  ((Answer  x))  ((VicePresident  x)))))  ;  Theorem  Clause
begins here
```

3.Obvously, two-pointer takes more steps than unit-preference. In unit-preference, every time I append a clause from the theorem clause list to the knowledge base clause list which is also when I pop the first clause of the theorem clause list, I sort all the left clauses in the theorem clause list. It’s more likely that the clause with less liberals can be resolved and derive false. And also, less-liberal clause can avoid more intermediate resovlents.But two-pointer never do the sorting step. That’s why there is an obvious difference between the two methods. Two-pointer has more unneceaasry intermediate resolution steps.

4. Full Resolution Steps:
(1)Howling Hound

```
CG-USER(120): (defvar *howling*      '((1 ((howl z)) ((hound z)))
                                   (2 nil ((have x y) (cat y) (have x z) (mouse z)))
                                   (3 nil ((ls w) (have w v) (howl v)))
                                   (4 ((have (john) (a))) nil)
                                   (5 ((cat (a)) (hound (a))) nil)
                                   (6 ((mouse (b))) nil)
                                   (7 ((ls (john))) nil)
                                   (8 ((have (john) (b))) nil)))

CG-USER(122): (two-pointer *howling* 6)

(1 ((HOWL Z)) ((HOUND Z)))
(2 NIL ((HAVE X Y) (CAT Y) (HAVE X Z) (MOUSE Z)))
(3 NIL ((LS W) (HAVE W V) (HOWL V)))
(4 ((HAVE (JOHN) (A))) NIL)
(5 ((CAT (A)) (HOUND (A))) NIL)
(6 ((MOUSE (B))) NIL)
(7 ((LS (JOHN))) NIL)
(8 ((HAVE (JOHN) (B))) NIL)
(9 NIL ((HAVE X Y) (CAT Y) (HAVE X (B))) 2 + 6)
(10 NIL ((HAVE (JOHN) V) (HOWL V)) 3 + 7)
(11 NIL ((CAT (B)) (HAVE (JOHN) Z) (MOUSE Z)) 2 + 8)
(12 NIL ((HAVE (JOHN) Y) (CAT Y) (MOUSE (B))) 2 + 8)
(13 NIL ((LS (JOHN)) (HOWL (B))) 3 + 8)
(14 NIL ((CAT (A)) (HAVE (JOHN) (B))) 4 + 9)
(15 ((HOUND (A))) ((HAVE X (A)) (HAVE X (B))) 5 + 9)
(16 NIL ((CAT (B))) 8 + 9)
(17 NIL ((HAVE (JOHN) Y) (CAT Y)) 8 + 9)
(18 NIL ((HAVE (JOHN) Z) (HOUND Z)) 1 + 10)
(19 NIL ((HOWL (A))) 4 + 10)
(20 NIL ((HOWL (B))) 8 + 10)
(21 NIL ((CAT (B)) (MOUSE (A))) 4 + 11)
(22 NIL ((CAT (B)) (HAVE (JOHN) (B))) 6 + 11)
(23 NIL ((CAT (B)) (MOUSE (B))) 8 + 11)
```

```
(24 NIL ((CAT (A)) (MOUSE (B))) 4 + 12)
(25 ((HOUND (A))) ((HAVE (JOHN) (A)) (MOUSE (B))) 5 + 12)
(26 NIL ((LS (JOHN)) (HOUND (B))) 1 + 13)
(27 ((HOUND (A))) ((HAVE (JOHN) (B))) 5 + 14)
(28 NIL ((CAT (A))) 8 + 14)
(29 ((HOWL (A))) ((HAVE X (A)) (HAVE X (B))) 1 + 15)
(30 ((HOUND (A))) ((HAVE (JOHN) (A))) 8 + 15)
(31 NIL ((HOUND (A))) 4 + 18)
(32 ((CAT (A))) ((HAVE (JOHN) (A))) 5 + 18)
(33 NIL ((HOUND (B))) 8 + 18)
(34 NIL ((HAVE (JOHN) (A)) (HAVE X (A)) (HAVE X (B))) 15 + 18)
(35 ((HOUND (A))) ((MOUSE (B))) 5 + 24)
(36 ((HOWL (A))) ((HAVE (JOHN) (A)) (MOUSE (B))) 1 + 25)
(37 NIL ((HAVE (JOHN) (A)) (HAVE (JOHN) (A)) (MOUSE (B))) 18 + 25)
(38 ((HOWL (A))) ((HAVE (JOHN) (B))) 1 + 27)
(39 ((HOUND (A))) NIL 8 + 27)
(40 NIL ((HAVE (JOHN) (A)) (HAVE (JOHN) (B))) 18 + 27)
(41 NIL ((LS W) (HAVE W (A)) (HAVE X (A)) (HAVE X (B))) 3 + 29)
(42 ((HOWL (A))) ((HAVE (JOHN) (A))) 8 + 29)
(43 NIL ((HAVE X (A)) (HAVE X (B))) 19 + 29)
(44 NIL ((HAVE (JOHN) (A)) (HAVE (JOHN) (A))) 18 + 30)
(45 ((CAT (A))) NIL 5 + 31)
(46 NIL ((HAVE (JOHN) (B))) 27 + 31)
(47 NIL ((HAVE (JOHN) (A))) 30 + 31)
(48 NIL ((HAVE X (A)) (HAVE X Z) (MOUSE Z) (HAVE (JOHN) (A))) 2 + 32)
(49 ((HOWL (A))) ((MOUSE (B))) 1 + 35)
(50 NIL ((MOUSE (B))) 31 + 35)
(51 NIL ((LS W) (HAVE W (A)) (HAVE (JOHN) (A)) (MOUSE (B))) 3 + 36)
(52 NIL ((LS W) (HAVE W (A)) (HAVE (JOHN) (B))) 3 + 38)
(53 ((HOWL (A))) NIL 8 + 38)
(31 + 39 = FALSE)
Theorem was successfully proven!

CG-USER(123): (unit-preference *howling* 6)

(1 ((HOWL Z)) ((HOUND Z)))
(2 NIL ((HAVE X Y) (CAT Y) (HAVE X Z) (MOUSE Z)))
(3 NIL ((LS W) (HAVE W V) (HOWL V)))
(4 ((HAVE (JOHN) (A))) NIL)
(5 ((CAT (A)) (HOUND (A))) NIL)
(6 ((MOUSE (B))) NIL)
(7 ((LS (JOHN))) NIL)
(8 ((HAVE (JOHN) (B))) NIL)
(9 NIL ((HAVE X Y) (CAT Y) (HAVE X (B))) 2 + 6)
```

```
(10 NIL ((HAVE (JOHN) V) (HOWL V)) 3 + 7)
(11 NIL ((CAT (B)) (HAVE (JOHN) Z) (MOUSE Z)) 2 + 8)
(12 NIL ((HAVE (JOHN) Y) (CAT Y) (MOUSE (B))) 2 + 8)
(13 NIL ((LS (JOHN)) (HOWL (B))) 3 + 8)
(14 NIL ((HAVE (JOHN) Z) (HOUND Z)) 1 + 10)
(15 NIL ((HOWL (A))) 4 + 10)
(16 NIL ((HOWL (B))) 8 + 10)
(17 NIL ((HOUND (A))) 1 + 15)
(18 NIL ((HOUND (B))) 1 + 16)
(19 ((CAT (A))) NIL 5 + 17)
(20 NIL ((HAVE X (A)) (HAVE X Z) (MOUSE Z)) 2 + 19)
(21 NIL ((LS (JOHN)) (HOUND (B))) 1 + 13)
(22 ((CAT (A))) ((HAVE (JOHN) (A))) 5 + 14)
(23 NIL ((HAVE X (A)) (HAVE X Z) (MOUSE Z) (HAVE (JOHN) (A))) 2 + 22)
(24 NIL ((CAT (A)) (HAVE (JOHN) (B))) 4 + 9)
(25 ((HOUND (A))) ((HAVE X (A)) (HAVE X (B))) 5 + 9)
(26 NIL ((CAT (B))) 8 + 9)
(27 NIL ((HAVE (JOHN) Y) (CAT Y)) 8 + 9)
(28 NIL ((HAVE X (A)) (HAVE X (B))) 19 + 9)
(29 NIL ((HAVE X (A)) (HAVE X (B)) (HAVE (JOHN) (A))) 22 + 9)
(30 ((HOUND (A))) ((HAVE (JOHN) (B))) 5 + 24)
(31 NIL ((CAT (A))) 8 + 24)
(32 NIL ((HAVE (JOHN) (B))) 19 + 24)
(33 NIL ((HAVE (JOHN) (B)) (HAVE (JOHN) (A))) 22 + 24)
(34 ((HOUND (A))) NIL 5 + 31)
(19 + 31 = FALSE)
Theorem was successfully proven!
```

(2) Coyote and roadrunner

```
CG-USER(124): (defvar *rr*   '( (1    ( (rr (a)) )          ( (coyote y) )    )
                   (2    ( (chase z (a)) ) ( (coyote z) )    )
                   (3    ( (smart x) )    ( (rr x) (beep x) )    )
                   (4    nil ( (coyote w) (rr u) (catch w u) (smart u) )      )
                   (5    ( (frustrated s) (catch s t) ) ( (coyote s) (rr t)
                          (chase s t) )      )
                   (6    ( (beep r) )   ( (rr r) )    )
                   (7    ( ( coyote (b)) ) nil    ) ;; goal clause begins here
                   (8    nil ( (frustrated (b)) )    )        ) )

CG-USER(125): (two-pointer *rr* 7)

(1 ((RR (A))) ((COYOTE Y)))
(2 ((CHASE Z (A))) ((COYOTE Z)))
(3 ((SMART X)) ((RR X) (BEEP X)))
```

```
(4 NIL ((COYOTE W) (RR U) (CATCH W U) (SMART U)))
(5 ((FRUSTRATED S) (CATCH S T)) ((COYOTE S) (RR T) (CHASE S T)))
(6 ((BEEP R)) ((RR R)))
(7 ((COYOTE (B))) NIL)
(8 NIL ((FRUSTRATED (B))))
(9 ((RR (A))) NIL 1 + 7)
(10 ((CHASE (B) (A))) NIL 2 + 7)
(11 NIL ((RR U) (CATCH (B) U) (SMART U)) 4 + 7)
(12 ((FRUSTRATED (B)) (CATCH (B) T)) ((RR T) (CHASE (B) T)) 5 + 7)
(13 ((CATCH (B) T)) ((COYOTE (B)) (RR T) (CHASE (B) T)) 5 + 8)
(14 ((SMART (A))) ((BEEP (A))) 3 + 9)
(15 NIL ((COYOTE W) (CATCH W (A)) (SMART (A))) 4 + 9)
(16 ((FRUSTRATED S) (CATCH S (A))) ((COYOTE S) (CHASE S (A))) 5 + 9)
(17 ((BEEP (A))) NIL 6 + 9)
(18 ((FRUSTRATED (B)) (CATCH (B) (A))) ((COYOTE (B)) (RR (A))) 5 + 10)
(19 NIL ((CATCH (B) (A)) (SMART (A)) (COYOTE Y)) 1 + 11)
(20 NIL ((RR X) (CATCH (B) X) (RR X) (BEEP X)) 3 + 11)
(21 ((FRUSTRATED (B))) ((RR T) (SMART T) (COYOTE (B)) (RR T) (CHASE (B) T)) 5 + 11)
(22 NIL ((CATCH (B) (A)) (SMART (A))) 9 + 11)
(23 ((FRUSTRATED (B)) (CATCH (B) (A))) ((CHASE (B) (A)) (COYOTE Y)) 1 + 12)
(24 ((FRUSTRATED (B))) ((COYOTE (B)) (RR U) (SMART U) (RR U) (CHASE (B) U)) 4 + 12)
(25 ((CATCH (B) T)) ((RR T) (CHASE (B) T)) 8 + 12)
(26 ((FRUSTRATED (B)) (CATCH (B) (A))) ((CHASE (B) (A))) 9 + 12)
(27 ((FRUSTRATED (B)) (CATCH (B) (A))) ((RR (A))) 10 + 12)
(28 ((FRUSTRATED (B))) ((RR U) (SMART U) (RR U) (CHASE (B) U)) 11 + 12)
(29 ((CATCH (B) (A))) ((COYOTE (B)) (CHASE (B) (A)) (COYOTE Y)) 1 + 13)
(30 ((CATCH (B) (A))) ((COYOTE (B)) (RR (A)) (COYOTE (B))) 2 + 13)
(31 NIL ((COYOTE (B)) (RR U) (SMART U) (COYOTE (B)) (RR U) (CHASE (B) U)) 4 + 13)
(32 ((CATCH (B) (A))) ((COYOTE (B)) (CHASE (B) (A))) 9 + 13)
(33 ((CATCH (B) (A))) ((COYOTE (B)) (RR (A))) 10 + 13)
(34 NIL ((RR U) (SMART U) (COYOTE (B)) (RR U) (CHASE (B) U)) 11 + 13)
(35 NIL ((COYOTE W) (RR (A)) (CATCH W (A)) (BEEP (A))) 4 + 14)
(36 ((SMART (A))) ((RR (A))) 6 + 14)
(37 NIL ((RR (A)) (CATCH (B) (A)) (BEEP (A))) 11 + 14)
(38 ((FRUSTRATED S)) ((COYOTE S) (SMART (A)) (COYOTE S) (RR (A)) (CHASE S (A))) 5 + 15)
(39 ((FRUSTRATED (B))) ((COYOTE (B)) (SMART (A)) (RR (A)) (CHASE (B) (A))) 12 + 15)
(40 NIL ((COYOTE (B)) (SMART (A)) (COYOTE (B)) (RR (A)) (CHASE (B) (A))) 13 + 15)
(41 NIL ((COYOTE W) (CATCH W (A)) (BEEP (A))) 14 + 15)
(42 ((FRUSTRATED Z) (CATCH Z (A))) ((COYOTE Z) (COYOTE Z)) 2 + 16)
(43 ((FRUSTRATED W)) ((COYOTE W) (RR (A)) (SMART (A)) (COYOTE W) (CHASE W (A))) 4 + 16)
(44 ((FRUSTRATED (B)) (CATCH (B) (A))) ((COYOTE (B))) 10 + 16)
(45 ((FRUSTRATED W)) ((COYOTE W) (SMART (A)) (COYOTE W) (CHASE W (A))) 15 + 16)
(46 ((SMART (A))) NIL 14 + 17)
(47 ((FRUSTRATED (B)) (CATCH (B) (A))) ((COYOTE (B)) (COYOTE Y)) 1 + 18)
```

```
(48 ((FRUSTRATED (B))) ((COYOTE (B)) (RR (A)) (SMART (A)) (COYOTE (B)) (RR (A))) 4 + 18)
(49 ((FRUSTRATED (B))) ((RR (A)) (SMART (A)) (COYOTE (B)) (RR (A))) 11 + 18)
(50 NIL ((CATCH (B) (A)) (COYOTE Y) (RR (A)) (BEEP (A))) 3 + 19)
(51 ((FRUSTRATED (B))) ((SMART (A)) (COYOTE Y) (RR (A)) (CHASE (B) (A))) 12 + 19)
(52 NIL ((CATCH (B) (A)) (COYOTE Y) (BEEP (A))) 14 + 19)
(53 ((FRUSTRATED (B))) ((SMART (A)) (COYOTE Y) (COYOTE (B)) (CHASE (B) (A))) 16 + 19)
(54 ((FRUSTRATED (B))) ((RR T) (RR T) (BEEP T) (COYOTE (B)) (RR T) (CHASE (B) T)) 5 + 20)
(55 NIL ((RR R) (CATCH (B) R) (RR R) (RR R)) 6 + 20)
(56 NIL ((CATCH (B) (A)) (BEEP (A))) 9 + 20)
(57 ((FRUSTRATED (B))) ((RR T) (RR T) (BEEP T) (RR T) (CHASE (B) T)) 12 + 20)
(58 NIL ((RR T) (RR T) (BEEP T) (COYOTE (B)) (RR T) (CHASE (B) T)) 13 + 20)
(59 ((FRUSTRATED (B))) ((RR (A)) (RR (A)) (BEEP (A)) (COYOTE (B)) (CHASE (B) (A))) 16 + 20)
(60 NIL ((RR (A)) (CATCH (B) (A)) (RR (A))) 17 + 20)
(61 ((FRUSTRATED (B))) ((RR (A)) (RR (A)) (BEEP (A)) (COYOTE (B)) (RR (A))) 18 + 20)
(62 ((FRUSTRATED (B))) ((RR X) (COYOTE (B)) (RR X) (CHASE (B) X) (RR X) (BEEP X)) 3 + 21)
(63 ((FRUSTRATED (B))) ((RR T) (SMART T) (RR T) (CHASE (B) T)) 7 + 21)
(64 NIL ((RR T) (SMART T) (COYOTE (B)) (RR T) (CHASE (B) T)) 8 + 21)
(65 ((FRUSTRATED (B))) ((SMART (A)) (COYOTE (B)) (CHASE (B) (A))) 9 + 21)
(66 ((FRUSTRATED (B))) ((SMART (A)) (RR (A)) (CHASE (B) (A))) 12 + 22)
(67 NIL ((SMART (A)) (COYOTE (B)) (RR (A)) (CHASE (B) (A))) 13 + 22)
(68 ((FRUSTRATED (B))) ((SMART (A)) (COYOTE (B)) (RR (A))) 18 + 22)
(69 ((CATCH (B) (A))) ((CHASE (B) (A)) (COYOTE Y)) 8 + 23)
(70 ((FRUSTRATED (B)) (CATCH (B) (A))) ((COYOTE Y)) 10 + 23)
(71 ((FRUSTRATED (B))) ((SMART (A)) (COYOTE Y) (CHASE (B) (A)) (COYOTE Y)) 19 + 23)
(72 ((FRUSTRATED (B))) ((RR (A)) (RR (A)) (BEEP (A)) (CHASE (B) (A)) (COYOTE Y)) 20 + 23)
(73 ((FRUSTRATED (B))) ((SMART (A)) (CHASE (B) (A)) (COYOTE Y)) 22 + 23)
(74 ((CATCH (B) (A))) ((CHASE (B) (A))) 9 + 25)
(75 ((CATCH (B) (A))) ((RR (A))) 10 + 25)
(76 NIL ((RR U) (SMART U) (RR U) (CHASE (B) U)) 11 + 25)
(77 NIL ((SMART (A)) (COYOTE Y) (RR (A)) (CHASE (B) (A))) 19 + 25)
(78 NIL ((RR X) (RR X) (BEEP X) (RR X) (CHASE (B) X)) 20 + 25)
(79 NIL ((SMART (A)) (RR (A)) (CHASE (B) (A))) 22 + 25)
(80 ((FRUSTRATED (B)) (CATCH (B) (A))) NIL 10 + 26)
(81 ((FRUSTRATED (B))) ((RR (A)) (RR (A)) (BEEP (A)) (CHASE (B) (A))) 20 + 26)
(82 ((FRUSTRATED (B))) ((SMART (A)) (CHASE (B) (A))) 22 + 26)
(83 ((FRUSTRATED (B))) ((RR (A)) (SMART (A)) (RR (A))) 11 + 27)
(84 ((FRUSTRATED (B))) ((RR (A)) (RR (A)) (BEEP (A)) (RR (A))) 20 + 27)
(85 ((FRUSTRATED (B))) ((SMART (A)) (RR (A))) 22 + 27)
(86 ((FRUSTRATED (B))) ((RR X) (RR X) (CHASE (B) X) (RR X) (BEEP X)) 3 + 28)
(87 ((CATCH (B) (A))) ((COYOTE (B)) (COYOTE Y) (COYOTE (B))) 2 + 29)
(88 NIL ((COYOTE (B)) (RR (A)) (SMART (A)) (COYOTE (B)) (CHASE (B) (A)) (COYOTE Y)) 4 + 29)
(89 ((CATCH (B) (A))) ((COYOTE (B)) (COYOTE Y)) 10 + 29)
(90 NIL ((COYOTE (B)) (SMART (A)) (COYOTE (B)) (CHASE (B) (A)) (COYOTE Y)) 15 + 29)
(91 NIL ((RR (A)) (RR (A)) (BEEP (A)) (COYOTE (B)) (CHASE (B) (A)) (COYOTE Y)) 20 + 29)
```

```
(92 NIL ((SMART (A)) (COYOTE (B)) (CHASE (B) (A)) (COYOTE Y)) 22 + 29)
(93 NIL ((COYOTE (B)) (RR (A)) (SMART (A)) (COYOTE (B)) (RR (A)) (COYOTE (B))) 4 + 30)
(94 ((CATCH (B) (A))) ((COYOTE (B)) (COYOTE (B))) 9 + 30)
(95 NIL ((RR (A)) (SMART (A)) (COYOTE (B)) (RR (A)) (COYOTE (B))) 11 + 30)
(96 NIL ((RR (A)) (RR (A)) (BEEP (A)) (COYOTE (B)) (RR (A)) (COYOTE (B))) 20 + 30)
(97 NIL ((SMART (A)) (COYOTE (B)) (RR (A)) (COYOTE (B))) 22 + 30)
(98 NIL ((COYOTE (B)) (RR X) (COYOTE (B)) (RR X) (CHASE (B) X) (RR X) (BEEP X)) 3 + 31)
(99 NIL ((COYOTE (B)) (SMART (A)) (COYOTE (B)) (CHASE (B) (A))) 9 + 31)
(100 ((CATCH (B) (A))) ((COYOTE (B))) 10 + 32)
(101 NIL ((RR (A)) (RR (A)) (BEEP (A)) (COYOTE (B)) (CHASE (B) (A))) 20 + 32)
(102 NIL ((SMART (A)) (COYOTE (B)) (CHASE (B) (A))) 22 + 32)
(103 NIL ((RR (A)) (RR (A)) (BEEP (A)) (COYOTE (B)) (RR (A))) 20 + 33)
(104 NIL ((SMART (A)) (COYOTE (B)) (RR (A))) 22 + 33)
(105 NIL ((RR X) (COYOTE (B)) (RR X) (CHASE (B) X) (RR X) (BEEP X)) 3 + 34)
(106 NIL ((COYOTE W) (CATCH W (A)) (BEEP (A)) (COYOTE Y)) 1 + 35)
(107 ((FRUSTRATED S)) ((COYOTE S) (RR (A)) (BEEP (A)) (COYOTE S) (RR (A)) (CHASE S (A))) 5 + 35)
(108 NIL ((COYOTE W) (RR (A)) (CATCH W (A)) (RR (A))) 6 + 35)
(109 ((FRUSTRATED S)) ((COYOTE S) (RR (A)) (BEEP (A)) (COYOTE S) (CHASE S (A))) 16 + 35)
(110 NIL ((COYOTE W) (RR (A)) (CATCH W (A))) 17 + 35)
(111 ((SMART (A))) ((COYOTE Y)) 1 + 36)
(112 ((FRUSTRATED (B))) ((RR (A)) (COYOTE (B)) (RR (A)) (CHASE (B) (A)) (RR (A))) 21 + 36)
(113 NIL ((CATCH (B) (A)) (RR (A))) 22 + 36)
(114 ((FRUSTRATED (B))) ((RR (A)) (RR (A)) (CHASE (B) (A)) (RR (A))) 28 + 36)
(115 NIL ((COYOTE (B)) (RR (A)) (COYOTE (B)) (RR (A)) (CHASE (B) (A)) (RR (A))) 31 + 36)
(116 NIL ((RR (A)) (COYOTE (B)) (RR (A)) (CHASE (B) (A)) (RR (A))) 34 + 36)
(117 NIL ((RR (A)) (BEEP (A)) (RR (A)) (CHASE (B) (A))) 25 + 37)
(118 ((FRUSTRATED (B))) ((RR (A)) (BEEP (A)) (CHASE (B) (A))) 26 + 37)
(119 ((FRUSTRATED (B))) ((RR (A)) (BEEP (A)) (RR (A))) 27 + 37)
(120 NIL ((RR (A)) (BEEP (A)) (COYOTE (B)) (RR (A))) 33 + 37)
(121 ((FRUSTRATED S)) ((COYOTE S) (SMART (A)) (COYOTE S) (CHASE S (A)) (COYOTE Y)) 1 + 38)
(122 ((FRUSTRATED Z)) ((COYOTE Z) (SMART (A)) (COYOTE Z) (RR (A)) (COYOTE Z)) 2 + 38)
(123 ((FRUSTRATED S)) ((COYOTE S) (SMART (A)) (COYOTE S) (CHASE S (A))) 9 + 38)
(124 ((FRUSTRATED S)) ((COYOTE S) (COYOTE S) (RR (A)) (CHASE S (A)) (RR (A))) 36 + 38)
(125 ((FRUSTRATED S)) ((COYOTE S) (BEEP (A)) (COYOTE S) (CHASE S (A))) 16 + 41)
(126 NIL ((COYOTE W) (CATCH W (A))) 17 + 41)
(127 ((FRUSTRATED (B))) ((COYOTE (B)) (BEEP (A)) (CHASE (B) (A)) (COYOTE Y)) 23 + 41)
(128 ((FRUSTRATED (B))) ((COYOTE (B)) (BEEP (A)) (CHASE (B) (A))) 26 + 41)
(129 NIL ((COYOTE (B)) (BEEP (A)) (COYOTE (B)) (CHASE (B) (A)) (COYOTE Y)) 29 + 41)
(130 NIL ((COYOTE (B)) (BEEP (A)) (COYOTE (B)) (CHASE (B) (A))) 32 + 41)
(131 ((FRUSTRATED W)) ((COYOTE W) (RR (A)) (SMART (A)) (COYOTE W) (COYOTE W)) 4 + 42)
(132 ((FRUSTRATED W)) ((COYOTE W) (SMART (A)) (COYOTE W) (COYOTE W)) 15 + 42)
(133 ((FRUSTRATED (B))) ((SMART (A)) (COYOTE Y) (COYOTE (B)) (COYOTE (B))) 19 + 42)
(134 ((FRUSTRATED (B))) ((SMART (A)) (COYOTE (B)) (COYOTE (B))) 22 + 42)
(135 ((FRUSTRATED W)) ((COYOTE W) (RR (A)) (BEEP (A)) (COYOTE W) (COYOTE W)) 35 + 42)
```

```
(136 ((FRUSTRATED W)) ((COYOTE W) (BEEP (A)) (COYOTE W) (COYOTE W)) 41 + 42)
(137 ((FRUSTRATED W)) ((COYOTE W) (SMART (A)) (COYOTE W) (CHASE W (A)) (COYOTE Y)) 1 + 43)
(138 ((FRUSTRATED W)) ((COYOTE W) (RR (A)) (COYOTE W) (CHASE W (A)) (RR (A)) (BEEP (A))) 3 + 43)
(139 ((FRUSTRATED W)) ((COYOTE W) (RR (A)) (COYOTE W) (CHASE W (A)) (RR (A))) 36 + 43)
(140 ((FRUSTRATED (B))) ((SMART (A)) (COYOTE (B))) 22 + 44)
(141 ((FRUSTRATED (B))) ((COYOTE (B)) (BEEP (A)) (COYOTE (B))) 41 + 44)
(142 ((FRUSTRATED Z)) ((COYOTE Z) (SMART (A)) (COYOTE Z) (COYOTE Z)) 2 + 45)
(143 ((FRUSTRATED W)) ((COYOTE W) (COYOTE W) (CHASE W (A)) (RR (A))) 36 + 45)
(144 NIL ((CATCH (B) (A)) (COYOTE Y)) 19 + 46)
(145 NIL ((CATCH (B) (A))) 22 + 46)
(146 ((FRUSTRATED (B))) ((RR (A)) (RR (A)) (CHASE (B) (A))) 28 + 46)
(147 NIL ((RR (A)) (COYOTE (B)) (RR (A)) (CHASE (B) (A))) 34 + 46)
(148 ((FRUSTRATED S)) ((COYOTE S) (COYOTE S) (RR (A)) (CHASE S (A))) 38 + 46)
(149 ((FRUSTRATED W)) ((COYOTE W) (COYOTE W) (CHASE W (A))) 45 + 46)
(150 ((FRUSTRATED (B))) ((COYOTE (B)) (BEEP (A)) (COYOTE (B)) (COYOTE Y)) 41 + 47)
(151 ((FRUSTRATED (B))) ((COYOTE (B)) (RR (A)) (COYOTE (B)) (RR (A)) (RR (A)) (BEEP (A))) 3 + 48)
(152 ((FRUSTRATED (B))) ((COYOTE (B)) (RR (A)) (COYOTE (B)) (RR (A)) (RR (A))) 36 + 48)
(153 ((FRUSTRATED (B))) ((COYOTE (B)) (RR (A)) (COYOTE (B)) (RR (A))) 46 + 48)
(154 ((FRUSTRATED (B))) ((RR (A)) (COYOTE (B)) (RR (A))) 46 + 49)
(155 NIL ((CATCH (B) (A)) (COYOTE Y) (BEEP (A)) (COYOTE Y)) 1 + 50)
(156 NIL ((CATCH (B) (A)) (COYOTE Y) (RR (A)) (RR (A))) 6 + 50)
(157 NIL ((COYOTE Y) (RR (A)) (BEEP (A)) (RR (A)) (CHASE (B) (A))) 25 + 50)
(158 ((FRUSTRATED (B))) ((COYOTE Y) (BEEP (A)) (CHASE (B) (A)) (COYOTE Y)) 23 + 52)
(159 ((FRUSTRATED (B))) ((COYOTE Y) (BEEP (A)) (CHASE (B) (A))) 26 + 52)
(160 ((FRUSTRATED (B))) ((COYOTE Y) (COYOTE (B)) (CHASE (B) (A))) 46 + 53)
(161 ((FRUSTRATED (B))) ((RR R) (RR R) (COYOTE (B)) (RR R) (CHASE (B) R) (RR R)) 6 + 54)
(162 ((FRUSTRATED (B))) ((RR T) (RR T) (RR T) (COYOTE (B)) (RR T) (CHASE (B) T)) 5 + 55)
(163 ((FRUSTRATED (B))) ((RR T) (RR T) (RR T) (RR T) (CHASE (B) T)) 12 + 55)
(164 NIL ((RR T) (RR T) (RR T) (COYOTE (B)) (RR T) (CHASE (B) T)) 13 + 55)
(165 ((FRUSTRATED (B))) ((RR (A)) (RR (A)) (RR (A)) (CHASE (B) (A)) (COYOTE Y)) 23 + 55)
(166 NIL ((RR T) (RR T) (RR T) (RR T) (CHASE (B) T)) 25 + 55)
(167 ((FRUSTRATED (B))) ((RR (A)) (RR (A)) (RR (A)) (RR (A))) 27 + 55)
(168 NIL ((RR (A)) (RR (A)) (RR (A)) (COYOTE (B)) (RR (A)) (COYOTE (B))) 30 + 55)
(169 NIL ((RR (A)) (RR (A)) (RR (A)) (COYOTE (B)) (RR (A))) 33 + 55)
(170 NIL ((BEEP (A)) (RR (A)) (CHASE (B) (A))) 25 + 56)
(171 ((FRUSTRATED (B))) ((BEEP (A)) (CHASE (B) (A))) 26 + 56)
(172 ((FRUSTRATED (B))) ((BEEP (A)) (RR (A))) 27 + 56)
(173 NIL ((BEEP (A)) (COYOTE (B)) (CHASE (B) (A))) 32 + 56)
(174 NIL ((BEEP (A)) (COYOTE (B)) (RR (A))) 33 + 56)
(175 ((FRUSTRATED (B))) ((BEEP (A)) (COYOTE (B))) 44 + 56)
(176 ((FRUSTRATED (B))) ((RR R) (RR R) (RR R) (CHASE (B) R) (RR R)) 6 + 57)
(177 NIL ((RR R) (RR R) (COYOTE (B)) (RR R) (CHASE (B) R) (RR R)) 6 + 58)
(178 NIL ((RR (A)) (RR (A)) (RR (A)) (CHASE (B) (A))) 25 + 60)
```

```
(179 ((FRUSTRATED (B))) ((RR (A)) (RR (A)) (RR (A))) 27 + 60)
(180 NIL ((RR (A)) (RR (A)) (COYOTE (B)) (RR (A))) 33 + 60)
(181 NIL ((RR T) (SMART T) (RR T) (CHASE (B) T)) 8 + 63)
(182 ((FRUSTRATED (B))) ((COYOTE (B)) (CHASE (B) (A))) 46 + 65)
(183 ((FRUSTRATED (B))) ((RR (A)) (CHASE (B) (A))) 46 + 66)
(184 NIL ((COYOTE (B)) (RR (A)) (CHASE (B) (A))) 46 + 67)
(185 ((FRUSTRATED (B))) ((COYOTE (B)) (RR (A))) 46 + 68)
(186 ((CATCH (B) (A))) ((COYOTE Y)) 10 + 69)
(187 NIL ((SMART (A)) (COYOTE Y) (CHASE (B) (A)) (COYOTE Y)) 19 + 69)
(188 NIL ((SMART (A)) (CHASE (B) (A)) (COYOTE Y)) 22 + 69)
(189 NIL ((COYOTE Y) (BEEP (A)) (CHASE (B) (A)) (COYOTE Y)) 52 + 69)
(190 NIL ((RR (A)) (RR (A)) (RR (A)) (CHASE (B) (A)) (COYOTE Y)) 55 + 69)
(191 NIL ((BEEP (A)) (CHASE (B) (A)) (COYOTE Y)) 56 + 69)
(192 ((FRUSTRATED (B))) ((SMART (A)) (COYOTE Y) (COYOTE Y)) 19 + 70)
(193 ((FRUSTRATED (B))) ((SMART (A)) (COYOTE Y)) 22 + 70)
(194 ((FRUSTRATED (B))) ((COYOTE Y) (BEEP (A)) (COYOTE Y)) 52 + 70)
(195 ((FRUSTRATED (B))) ((BEEP (A)) (COYOTE Y)) 56 + 70)
(196 ((FRUSTRATED (B))) ((COYOTE Y) (CHASE (B) (A)) (COYOTE Y)) 46 + 71)
(197 ((FRUSTRATED (B))) ((CHASE (B) (A)) (COYOTE Y)) 46 + 73)
(198 ((CATCH (B) (A))) NIL 10 + 74)
(199 NIL ((SMART (A)) (CHASE (B) (A))) 22 + 74)
(200 NIL ((BEEP (A)) (CHASE (B) (A))) 56 + 74)
(201 NIL ((RR (A)) (RR (A)) (CHASE (B) (A))) 60 + 74)
(202 NIL ((RR (A)) (SMART (A)) (RR (A))) 11 + 75)
(203 NIL ((RR (A)) (RR (A)) (BEEP (A)) (RR (A))) 20 + 75)
(204 NIL ((SMART (A)) (RR (A))) 22 + 75)
(205 NIL ((RR (A)) (BEEP (A)) (RR (A))) 37 + 75)
(206 NIL ((RR (A)) (RR (A)) (RR (A)) (RR (A))) 55 + 75)
(207 NIL ((BEEP (A)) (RR (A))) 56 + 75)
(208 NIL ((RR (A)) (RR (A)) (RR (A))) 60 + 75)
(209 NIL ((RR R) (RR R) (RR R) (CHASE (B) R) (RR R)) 6 + 78)
(210 NIL ((RR (A)) (CHASE (B) (A))) 46 + 79)
(211 ((FRUSTRATED (B))) ((SMART (A))) 22 + 80)
(212 ((FRUSTRATED (B))) ((BEEP (A))) 56 + 80)
(213 ((FRUSTRATED (B))) ((RR (A)) (RR (A))) 60 + 80)
(214 ((FRUSTRATED (B))) ((CHASE (B) (A))) 46 + 82)
(215 ((FRUSTRATED (B))) ((RR (A))) 46 + 85)
(216 NIL ((COYOTE (B)) (SMART (A)) (COYOTE (B)) (COYOTE Y) (COYOTE (B))) 15 + 87)
(217 NIL ((SMART (A)) (COYOTE (B)) (COYOTE Y) (COYOTE (B))) 22 + 87)
(218 NIL ((COYOTE (B)) (BEEP (A)) (COYOTE (B)) (COYOTE Y) (COYOTE (B))) 41 + 87)
(219 NIL ((BEEP (A)) (COYOTE (B)) (COYOTE Y) (COYOTE (B))) 56 + 87)
(220 NIL ((COYOTE (B)) (SMART (A)) (COYOTE (B)) (CHASE (B) (A)) (COYOTE Y) (COYOTE Y)) 1 + 88)
(221 NIL ((COYOTE (B)) (RR (A)) (COYOTE (B)) (CHASE (B) (A)) (COYOTE Y) (RR (A)) (BEEP (A))) 3 + 88)
(222 NIL ((SMART (A)) (COYOTE (B)) (COYOTE Y)) 22 + 89)
```

```
(223 NIL ((BEEP (A)) (COYOTE (B)) (COYOTE Y)) 56 + 89)
(224 NIL ((COYOTE (B)) (COYOTE (B)) (CHASE (B) (A)) (COYOTE Y)) 46 + 90)
(225 NIL ((COYOTE (B)) (CHASE (B) (A)) (COYOTE Y)) 46 + 92)
(226 NIL ((COYOTE (B)) (RR (A)) (COYOTE (B)) (RR (A)) (COYOTE (B)) (RR (A)) (BEEP (A))) 3 + 93)
(227 NIL ((COYOTE (B)) (SMART (A)) (COYOTE (B)) (COYOTE (B))) 9 + 93)
(228 NIL ((SMART (A)) (COYOTE (B)) (COYOTE (B))) 22 + 94)
(229 NIL ((COYOTE (B)) (BEEP (A)) (COYOTE (B)) (COYOTE (B))) 41 + 94)
(230 NIL ((BEEP (A)) (COYOTE (B)) (COYOTE (B))) 56 + 94)
(231 NIL ((COYOTE (B)) (RR R) (COYOTE (B)) (RR R) (CHASE (B) R) (RR R) (RR R)) 6 + 98)
(232 NIL ((COYOTE (B)) (COYOTE (B)) (CHASE (B) (A))) 46 + 99)
(233 NIL ((SMART (A)) (COYOTE (B))) 22 + 100)
(234 NIL ((BEEP (A)) (COYOTE (B))) 56 + 100)
(235 NIL ((COYOTE (B)) (CHASE (B) (A))) 46 + 102)
(236 NIL ((COYOTE (B)) (RR (A))) 46 + 104)
(237 ((FRUSTRATED S)) ((COYOTE S) (BEEP (A)) (COYOTE Y) (COYOTE S) (CHASE S (A))) 16 + 106)
(238 NIL ((COYOTE W) (CATCH W (A)) (COYOTE Y)) 17 + 106)
(239 ((FRUSTRATED (B))) ((COYOTE (B)) (BEEP (A)) (COYOTE Y) (CHASE (B) (A)) (COYOTE Y)) 23 + 106)
(240 NIL ((COYOTE (B)) (BEEP (A)) (COYOTE Y) (COYOTE (B)) (CHASE (B) (A)) (COYOTE Y)) 29 + 106)
(241 ((FRUSTRATED Z)) ((COYOTE Z) (BEEP (A)) (COYOTE Y) (COYOTE Z) (COYOTE Z)) 42 + 106)
(242 ((FRUSTRATED (B))) ((COYOTE (B)) (BEEP (A)) (COYOTE Y) (COYOTE (B)) (COYOTE Y)) 47 + 106)
(243 NIL ((COYOTE (B)) (BEEP (A)) (COYOTE Y) (COYOTE (B)) (COYOTE Y) (COYOTE (B))) 87 + 106)
(244 ((FRUSTRATED Z)) ((COYOTE Z) (RR (A)) (BEEP (A)) (COYOTE Z) (RR (A)) (COYOTE Z)) 2 + 107)
(245 ((FRUSTRATED S)) ((COYOTE S) (RR (A)) (COYOTE S) (RR (A)) (CHASE S (A)) (RR (A))) 6 + 107)
(246 ((FRUSTRATED Z)) ((COYOTE Z) (RR (A)) (RR (A)) (COYOTE Z) (COYOTE Z)) 42 + 108)
(247 ((FRUSTRATED Z)) ((COYOTE Z) (RR (A)) (COYOTE Z) (COYOTE Z)) 42 + 110)
(248 NIL ((CATCH (B) (A)) (COYOTE Y) (COYOTE Y)) 19 + 111)
(249 ((FRUSTRATED W)) ((COYOTE W) (COYOTE W) (CHASE W (A)) (COYOTE Y)) 45 + 111)
(250 ((FRUSTRATED (B))) ((COYOTE Y) (COYOTE (B)) (CHASE (B) (A)) (COYOTE Y)) 53 + 111)
(251 ((FRUSTRATED (B))) ((COYOTE Y) (CHASE (B) (A)) (COYOTE Y) (COYOTE Y)) 71 + 111)
(252 NIL ((COYOTE (B)) (COYOTE (B)) (CHASE (B) (A)) (COYOTE Y) (COYOTE Y)) 90 + 111)
(253 NIL ((RR (A)) (RR (A))) 75 + 113)
(254 ((FRUSTRATED Z)) ((COYOTE Z) (SMART (A)) (COYOTE Z) (COYOTE Y) (COYOTE Z)) 2 + 121)
(255 ((FRUSTRATED S)) ((COYOTE S) (COYOTE S) (CHASE S (A)) (COYOTE Y)) 46 + 121)
(256 ((FRUSTRATED S)) ((COYOTE S) (COYOTE S) (CHASE S (A)) (COYOTE Y) (COYOTE Y)) 111 + 121)
(257 ((FRUSTRATED S)) ((COYOTE S) (COYOTE S) (CHASE S (A))) 46 + 123)
(258 ((FRUSTRATED Z)) ((COYOTE Z) (BEEP (A)) (COYOTE Z) (COYOTE Z)) 2 + 125)
(259 ((FRUSTRATED Z)) ((COYOTE Z) (COYOTE Z) (COYOTE Z)) 42 + 126)
(260 ((FRUSTRATED (B))) ((COYOTE (B)) (COYOTE (B))) 44 + 126)
(261 ((FRUSTRATED (B))) ((COYOTE (B)) (COYOTE (B)) (COYOTE Y)) 47 + 126)
(262 ((FRUSTRATED (B))) ((COYOTE (B))) 80 + 126)
(263 NIL ((COYOTE (B)) (COYOTE (B)) (COYOTE Y) (COYOTE (B))) 87 + 126)
(264 NIL ((COYOTE (B)) (COYOTE (B)) (COYOTE Y)) 89 + 126)
(265 NIL ((COYOTE (B)) (COYOTE (B)) (COYOTE (B))) 94 + 126)
(266 NIL ((COYOTE (B)) (COYOTE (B))) 100 + 126)
```

```
(267 ((FRUSTRATED W)) ((COYOTE W) (SMART (A)) (COYOTE W) (COYOTE W) (COYOTE Y)) 1 + 131)
(268 ((FRUSTRATED W)) ((COYOTE W) (RR (A)) (COYOTE W) (COYOTE W) (RR (A)) (BEEP (A))) 3 + 131)
(269 ((FRUSTRATED W)) ((COYOTE W) (RR (A)) (COYOTE W) (COYOTE W) (RR (A))) 36 + 131)
(270 ((FRUSTRATED W)) ((COYOTE W) (RR (A)) (COYOTE W) (COYOTE W)) 46 + 131)
(271 ((FRUSTRATED W)) ((COYOTE W) (COYOTE W) (COYOTE W)) 46 + 132)
(272 ((FRUSTRATED W)) ((COYOTE W) (COYOTE W) (COYOTE W) (COYOTE Y)) 111 + 132)
(273 ((FRUSTRATED (B))) ((COYOTE Y) (COYOTE (B)) (COYOTE (B)) (COYOTE Y)) 111 + 133)
(274 ((FRUSTRATED W)) ((COYOTE W) (BEEP (A)) (COYOTE W) (COYOTE W) (COYOTE Y)) 1 + 135)
(275 ((FRUSTRATED W)) ((COYOTE W) (COYOTE W) (CHASE W (A)) (COYOTE Y) (COYOTE Y)) 111 +
137)
(276 ((FRUSTRATED W)) ((COYOTE W) (RR (A)) (COYOTE W) (CHASE W (A)) (RR (A)) (RR (A))) 6 + 138)
(277 ((FRUSTRATED Z)) ((COYOTE Z) (COYOTE Z) (COYOTE Z) (COYOTE Y)) 111 + 142)
(278 NIL ((COYOTE Y) (CHASE (B) (A)) (COYOTE Y)) 69 + 144)
(279 ((FRUSTRATED (B))) ((COYOTE Y) (COYOTE Y)) 70 + 144)
(280 NIL ((COYOTE Y) (CHASE (B) (A))) 74 + 144)
(281 ((FRUSTRATED (B))) ((COYOTE Y)) 80 + 144)
(282 NIL ((CHASE (B) (A))) 74 + 145)
(283 NIL ((RR (A))) 75 + 145)
(284 ((FRUSTRATED (B))) NIL 80 + 145)
(285 NIL ((COYOTE (B))) 100 + 145)
(286 ((FRUSTRATED (B))) ((COYOTE (B)) (RR (A)) (COYOTE (B)) (RR (A)) (RR (A)) (RR (A))) 6 + 151)
(287 ((FRUSTRATED (B))) ((COYOTE Y) (BEEP (A)) (COYOTE Y) (CHASE (B) (A)) (COYOTE Y)) 23 + 155)
(288 NIL ((COYOTE Y) (BEEP (A)) (COYOTE Y) (CHASE (B) (A)) (COYOTE Y)) 69 + 155)
(289 ((FRUSTRATED (B))) ((COYOTE Y) (BEEP (A)) (COYOTE Y) (COYOTE Y)) 70 + 155)
(290 NIL ((SMART (A)) (COYOTE Y) (COYOTE Y)) 19 + 186)
(291 NIL ((SMART (A)) (COYOTE Y)) 22 + 186)
(292 NIL ((COYOTE Y) (BEEP (A)) (COYOTE Y)) 52 + 186)
(293 NIL ((BEEP (A)) (COYOTE Y)) 56 + 186)
(294 NIL ((COYOTE Y) (COYOTE Y)) 144 + 186)
(295 NIL ((COYOTE Y)) 145 + 186)
(296 NIL ((COYOTE Y) (BEEP (A)) (COYOTE Y) (COYOTE Y)) 155 + 186)
(297 NIL ((COYOTE Y) (CHASE (B) (A)) (COYOTE Y) (COYOTE Y)) 111 + 187)
(298 ((FRUSTRATED (B))) ((COYOTE Y) (COYOTE Y) (COYOTE Y)) 111 + 192)
(299 NIL ((SMART (A))) 22 + 198)
(300 NIL ((BEEP (A))) 56 + 198)
(145 + 198 = FALSE)
Theorem was successfully proven!

CG-USER(126): (unit-preference *rr* 7)

(1 ((RR (A))) ((COYOTE Y)))
(2 ((CHASE Z (A))) ((COYOTE Z)))
(3 ((SMART X)) ((RR X) (BEEP X)))
(4 NIL ((COYOTE W) (RR U) (CATCH W U) (SMART U)))
```

```
(5 ((FRUSTRATED S) (CATCH S T)) ((COYOTE S) (RR T) (CHASE S T)))
(6 ((BEEP R)) ((RR R)))
(7 ((COYOTE (B))) NIL)
(8 NIL ((FRUSTRATED (B))))
(9 ((RR (A))) NIL 1 + 7)
(10 ((CHASE (B) (A))) NIL 2 + 7)
(11 NIL ((RR U) (CATCH (B) U) (SMART U)) 4 + 7)
(12 ((FRUSTRATED (B)) (CATCH (B) T)) ((RR T) (CHASE (B) T)) 5 + 7)
(13 ((CATCH (B) T)) ((COYOTE (B)) (RR T) (CHASE (B) T)) 5 + 8)
(14 ((SMART (A))) ((BEEP (A))) 3 + 9)
(15 NIL ((COYOTE W) (CATCH W (A)) (SMART (A))) 4 + 9)
(16 ((FRUSTRATED S) (CATCH S (A))) ((COYOTE S) (CHASE S (A))) 5 + 9)
(17 ((BEEP (A))) NIL 6 + 9)
(18 ((FRUSTRATED (B)) (CATCH (B) (A))) ((COYOTE (B)) (RR (A))) 5 + 10)
(19 ((SMART (A))) ((RR (A))) 3 + 17)
(20 NIL ((COYOTE W) (RR (A)) (CATCH W (A)) (BEEP (A))) 4 + 14)
(21 ((SMART (A))) NIL 17 + 14)
(22 NIL ((COYOTE W) (RR (A)) (CATCH W (A))) 4 + 21)
(23 ((SMART (A))) ((COYOTE Y)) 1 + 19)
(24 NIL ((COYOTE W) (RR (A)) (CATCH W (A)) (RR (A))) 4 + 19)
(25 NIL ((CATCH (B) (A)) (SMART (A)) (COYOTE Y)) 1 + 11)
(26 NIL ((RR X) (CATCH (B) X) (RR X) (BEEP X)) 3 + 11)
(27 ((FRUSTRATED (B))) ((RR T) (SMART T) (COYOTE (B)) (RR T) (CHASE (B) T)) 5 + 11)
(28 NIL ((CATCH (B) (A)) (SMART (A))) 9 + 11)
(29 NIL ((RR (A)) (CATCH (B) (A)) (BEEP (A))) 14 + 11)
(30 NIL ((RR (A)) (CATCH (B) (A))) 21 + 11)
(31 NIL ((RR (A)) (CATCH (B) (A)) (RR (A))) 19 + 11)
(32 ((FRUSTRATED (B))) ((SMART (A)) (COYOTE (B)) (RR (A)) (CHASE (B) (A))) 5 + 28)
(33 NIL ((CATCH (B) (A)) (BEEP (A))) 14 + 28)
(34 NIL ((CATCH (B) (A))) 21 + 28)
(35 NIL ((CATCH (B) (A)) (COYOTE Y)) 23 + 28)
(36 ((FRUSTRATED (B))) ((COYOTE (B)) (RR (A)) (CHASE (B) (A))) 5 + 34)
(37 ((FRUSTRATED (B))) ((RR (A)) (COYOTE (B)) (RR (A)) (CHASE (B) (A))) 5 + 30)
(38 ((FRUSTRATED S)) ((COYOTE S) (SMART (A)) (COYOTE S) (RR (A)) (CHASE S (A))) 5 + 15)
(39 NIL ((COYOTE W) (CATCH W (A)) (BEEP (A))) 14 + 15)
(40 NIL ((COYOTE W) (CATCH W (A))) 21 + 15)
(41 NIL ((COYOTE W) (CATCH W (A)) (COYOTE Y)) 23 + 15)
(42 ((FRUSTRATED S)) ((COYOTE S) (COYOTE S) (RR (A)) (CHASE S (A))) 5 + 40)
(43 ((FRUSTRATED S)) ((COYOTE S) (RR (A)) (COYOTE S) (RR (A)) (CHASE S (A))) 5 + 22)
(44 NIL ((CATCH (B) (A)) (COYOTE Y) (RR (A)) (BEEP (A))) 3 + 25)
(45 ((FRUSTRATED (B))) ((SMART (A)) (COYOTE Y) (COYOTE (B)) (RR (A)) (CHASE (B) (A))) 5 + 25)
(46 NIL ((CATCH (B) (A)) (COYOTE Y) (BEEP (A))) 14 + 25)
(47 NIL ((CATCH (B) (A)) (COYOTE Y) (COYOTE Y)) 23 + 25)
(48 ((FRUSTRATED (B))) ((RR (A)) (BEEP (A)) (COYOTE (B)) (RR (A)) (CHASE (B) (A))) 5 + 29)
```

```
(49 ((FRUSTRATED (B))) ((RR (A)) (RR (A)) (COYOTE (B)) (RR (A)) (CHASE (B) (A))) 5 + 31)
(50 ((FRUSTRATED (B)) (CATCH (B) (A))) ((CHASE (B) (A)) (COYOTE Y)) 1 + 12)
(51 ((FRUSTRATED (B))) ((COYOTE (B)) (RR U) (SMART U) (RR U) (CHASE (B) U)) 4 + 12)
(52 ((CATCH (B) T)) ((RR T) (CHASE (B) T)) 8 + 12)
(53 ((FRUSTRATED (B)) (CATCH (B) (A))) ((CHASE (B) (A))) 9 + 12)
(54 ((FRUSTRATED (B)) (CATCH (B) (A))) ((RR (A))) 10 + 12)
(55 ((FRUSTRATED (B))) ((RR U) (SMART U) (RR U) (CHASE (B) U)) 11 + 12)
(56 ((FRUSTRATED (B))) ((SMART (A)) (RR (A)) (CHASE (B) (A))) 28 + 12)
(57 ((FRUSTRATED (B))) ((RR (A)) (CHASE (B) (A))) 34 + 12)
(58 ((FRUSTRATED (B))) ((RR (A)) (RR (A)) (CHASE (B) (A))) 30 + 12)
(59 ((FRUSTRATED (B))) ((SMART (A)) (COYOTE Y) (RR (A)) (CHASE (B) (A))) 25 + 12)
(60 ((FRUSTRATED (B))) ((RR (A)) (BEEP (A)) (RR (A)) (CHASE (B) (A))) 29 + 12)
(61 ((FRUSTRATED (B))) ((RR (A)) (RR (A)) (RR (A)) (CHASE (B) (A))) 31 + 12)
(62 ((CATCH (B) (A))) ((CHASE (B) (A)) (COYOTE Y)) 1 + 52)
(63 ((CATCH (B) (A))) ((RR (A)) (COYOTE (B))) 2 + 52)
(64 NIL ((COYOTE (B)) (RR U) (SMART U) (RR U) (CHASE (B) U)) 4 + 52)
(65 ((CATCH (B) (A))) ((CHASE (B) (A))) 9 + 52)
(66 ((CATCH (B) (A))) ((RR (A))) 10 + 52)
(67 NIL ((RR U) (SMART U) (RR U) (CHASE (B) U)) 11 + 52)
(68 NIL ((SMART (A)) (RR (A)) (CHASE (B) (A))) 28 + 52)
(69 NIL ((RR (A)) (CHASE (B) (A))) 34 + 52)
(70 NIL ((RR (A)) (RR (A)) (CHASE (B) (A))) 30 + 52)
(71 NIL ((COYOTE (B)) (SMART (A)) (RR (A)) (CHASE (B) (A))) 15 + 52)
(72 NIL ((COYOTE (B)) (RR (A)) (RR (A)) (CHASE (B) (A))) 22 + 52)
(73 NIL ((SMART (A)) (COYOTE Y) (RR (A)) (CHASE (B) (A))) 25 + 52)
(74 NIL ((RR (A)) (BEEP (A)) (RR (A)) (CHASE (B) (A))) 29 + 52)
(75 NIL ((RR (A)) (RR (A)) (RR (A)) (CHASE (B) (A))) 31 + 52)
(76 ((CATCH (B) (A))) ((COYOTE (B))) 2 + 65)
(77 ((CATCH (B) (A))) NIL 10 + 65)
(78 NIL ((SMART (A)) (CHASE (B) (A))) 28 + 65)
(79 NIL ((CHASE (B) (A))) 34 + 65)
(80 NIL ((BEEP (A)) (CHASE (B) (A))) 33 + 65)
(81 NIL ((COYOTE Y) (CHASE (B) (A))) 35 + 65)
(82 NIL ((COYOTE (B)) (SMART (A)) (CHASE (B) (A))) 15 + 65)
(83 NIL ((COYOTE (B)) (CHASE (B) (A))) 40 + 65)
(84 NIL ((SMART (A)) (COYOTE Y) (CHASE (B) (A))) 25 + 65)
(85 NIL ((COYOTE (B)) (BEEP (A)) (CHASE (B) (A))) 39 + 65)
(86 NIL ((COYOTE (B)) (COYOTE Y) (CHASE (B) (A))) 41 + 65)
(87 NIL ((COYOTE Y) (BEEP (A)) (CHASE (B) (A))) 46 + 65)
(88 NIL ((COYOTE Y) (COYOTE Y) (CHASE (B) (A))) 47 + 65)
(89 NIL ((COYOTE (B)) (RR (A)) (SMART (A))) 4 + 77)
(90 NIL ((RR (A)) (SMART (A))) 11 + 77)
(91 NIL ((SMART (A))) 28 + 77)
(34 + 77 = FALSE)
```

```
Theorem was successfully proven!
```

(3) Drug dealer and customs official

```
CG-USER(127): (defvar *customs* '( (1 ( (v x) (s x (f x)) )
                  ( (e x) ) )
              (2 ( (v y) (c (f y)) )
                  ( (e y) ) )
              (3 ( (e (a)) ) nil )
              (4 ( (d (a)) ) nil )
              (5 ( (d z) )      ( (s (a) z) ) )
              (6 nil      ( (d w) (v w) ) )
              (7 nil    ( (d r) (c r) ) )      )        ) ;; goal clause begins here
CG-USER(128): (two-pointer *CUSTOMS* 7)

(1 ((V X) (S X (F X))) ((E X)))
(2 ((V Y) (C (F Y))) ((E Y)))
(3 ((E (A))) NIL)
(4 ((D (A))) NIL)
(5 ((D Z)) ((S (A) Z)))
(6 NIL ((D W) (V W)))
(7 NIL ((D R) (C R)))
(8 ((V Y)) ((D (F Y)) (E Y)) 2 + 7)
(9 NIL ((C (A))) 4 + 7)
(10 NIL ((C Z) (S (A) Z)) 5 + 7)
(11 ((V (A))) ((D (F (A)))) 3 + 8)
(12 ((V Y)) ((E Y) (S (A) (F Y))) 5 + 8)
(13 NIL ((D W) (D (F W)) (E W)) 6 + 8)
(14 ((V (A))) ((C (F (A))) (E (A))) 1 + 10)
(15 ((V (A))) ((S (A) (F (A)))) 5 + 11)
(16 NIL ((D (A)) (D (F (A)))) 6 + 11)
(17 ((V (A)) (V (A))) ((E (A)) (E (A))) 1 + 12)
(18 NIL ((D W) (E W) (S (A) (F W))) 6 + 12)
(19 NIL ((D (F (A))) (E (A))) 4 + 13)
(20 NIL ((D (F Z)) (E Z) (S (A) Z)) 5 + 13)
(21 ((V (A))) ((C (F (A)))) 3 + 14)
(22 NIL ((D (A)) (C (F (A))) (E (A))) 6 + 14)
(23 ((V (A)) (V (A))) ((E (A))) 1 + 15)
(24 NIL ((D (A)) (S (A) (F (A)))) 6 + 15)
(25 NIL ((D (F (A)))) 4 + 16)
(26 NIL ((D (F (A))) (S (A) (A))) 5 + 16)
(27 ((V (A)) (V (A))) NIL 3 + 17)
(28 NIL ((D (A)) (E (A)) (E (A))) 6 + 17)
(29 ((V (A))) ((D (A)) (E (A)) (E (A))) 1 + 18)
(30 NIL ((E (A)) (S (A) (F (A)))) 4 + 18)
```

```
(31 NIL ((E Z) (S (A) (F Z)) (S (A) Z)) 5 + 18)
(32 ((V (A))) ((D (F (F (A)))) (E (F (A))) (E (A))) 1 + 20)
(33 NIL ((D (A)) (C (F (A)))) 6 + 21)
(34 NIL ((C (F (A))) (E (A))) 4 + 22)
(35 NIL ((C (F (A))) (E (A)) (S (A) (A))) 5 + 22)
(36 NIL ((D (A)) (E (A))) 6 + 23)
(37 ((V (A))) ((D (A)) (E (A))) 1 + 24)
(38 NIL ((S (A) (F (A)))) 4 + 24)
(39 NIL ((S (A) (F (A))) (S (A) (A))) 5 + 24)
(40 NIL ((D (A))) 6 + 27)
(41 NIL ((E (A)) (E (A))) 4 + 28)
(42 NIL ((E (A)) (E (A)) (S (A) (A))) 5 + 28)
(43 ((V (A))) ((D (A))) 3 + 29)
(44 ((V (A))) ((E (A)) (E (A))) 4 + 29)
(45 ((V (A))) ((E (A)) (E (A)) (S (A) (A))) 5 + 29)
(46 NIL ((D (A)) (D (A)) (E (A)) (E (A))) 6 + 29)
(47 ((V (A))) ((E (F (A))) (S (A) (F (F (A)))) (E (A))) 1 + 31)
(48 ((V (A))) ((D (F (F (A)))) (E (F (A)))) 3 + 32)
(49 NIL ((C (F (A)))) 4 + 33)
(50 NIL ((C (F (A))) (S (A) (A))) 5 + 33)
(51 NIL ((E (A))) 4 + 36)
(52 ((V (A))) ((E (A))) 4 + 37)
(4 + 40 = FALSE)
Theorem was successfully proven!

CG-USER(129): (unit-preference *CUSTOMS* 7)

(1 ((V X) (S X (F X))) ((E X)))
(2 ((V Y) (C (F Y))) ((E Y)))
(3 ((E (A))) NIL)
(4 ((D (A))) NIL)
(5 ((D Z)) ((S (A) Z)))
(6 NIL ((D W) (V W)))
(7 NIL ((D R) (C R)))
(8 ((V Y)) ((D (F Y)) (E Y)) 2 + 7)
(9 NIL ((C (A))) 4 + 7)
(10 NIL ((C Z) (S (A) Z)) 5 + 7)
(11 ((V (A))) ((C (F (A))) (E (A))) 1 + 10)
(12 ((V Y)) ((S (A) (F Y)) (E Y)) 2 + 10)
(13 ((V (A))) ((D (F (A)))) 3 + 8)
(14 NIL ((D W) (D (F W)) (E W)) 6 + 8)
(15 ((V (A))) ((S (A) (F (A)))) 5 + 13)
(16 NIL ((D (A)) (D (F (A)))) 6 + 13)
(17 ((V (A)) (V (A))) ((E (A))) 1 + 15)
```

```
(18 NIL ((D (A)) (S (A) (F (A)))) 6 + 15)
(19 NIL ((D (F (A)))) 4 + 16)
(20 NIL ((D (F (A))) (S (A) (A))) 5 + 16)
(21 NIL ((S (A) (F (A)))) 5 + 19)
(22 ((V (A))) ((E (A))) 1 + 21)
(23 ((V (A))) ((D (A)) (E (A))) 1 + 18)
(24 NIL ((S (A) (F (A))) (S (A) (A))) 5 + 18)
(25 ((V (A))) NIL 3 + 22)
(26 NIL ((D (A)) (E (A))) 6 + 22)
(27 NIL ((D (A))) 6 + 25)
(4 + 27 = FALSE)
Theorem was successfully proven!
```

(4) Harmonia (without Answer(x)):

```
(defvar *Harmonia* '((1 ((Grandparent x y)) ((Parent x z) (Parent z y)))
                        (2 ((Parent x y)) ((Mother x y))) (3 ((Parent x y)) ((Father x y))) (4 ((Father (Zeus)
(Ares))) nil)
                        (5 ((Mother (Hera) (Ares))) nil) (6 ((Father (Ares) (Harmonia))) nil)
                        (7 nil ((Grandparent x (Harmonia))))))
```

```
CG-USER(131): (two-pointer *HARMONIA* 7)

(1 ((GRANDPARENT X Y)) ((PARENT X Z) (PARENT Z Y)))
(2 ((PARENT X Y)) ((MOTHER X Y)))
(3 ((PARENT X Y)) ((FATHER X Y)))
(4 ((FATHER (ZEUS) (ARES))) NIL)
(5 ((MOTHER (HERA) (ARES))) NIL)
(6 ((FATHER (ARES) (HARMONIA))) NIL)
(7 NIL ((GRANDPARENT X (HARMONIA))))
(8 NIL ((PARENT X Z) (PARENT Z (HARMONIA))) 1 + 7)
(9 NIL ((PARENT Y (HARMONIA)) (MOTHER X Y)) 2 + 8)
(10 NIL ((PARENT X X) (MOTHER X (HARMONIA))) 2 + 8)
(11 NIL ((PARENT Y (HARMONIA)) (FATHER X Y)) 3 + 8)
(12 NIL ((PARENT X X) (FATHER X (HARMONIA))) 3 + 8)
(13 NIL ((MOTHER (HARMONIA) (HARMONIA)) (MOTHER (HARMONIA) (HARMONIA))) 2 + 9)
(14 NIL ((PARENT (ARES) (HARMONIA))) 5 + 9)
(15 NIL ((MOTHER Y (HARMONIA)) (MOTHER Y Y)) 2 + 10)
(16 NIL ((MOTHER Y (HARMONIA)) (FATHER Y Y)) 3 + 10)
(17 NIL ((FATHER (HARMONIA) (HARMONIA)) (FATHER (HARMONIA) (HARMONIA))) 3 + 11)
(18 NIL ((PARENT (HARMONIA) (HARMONIA))) 6 + 11)
(19 NIL ((FATHER Y (HARMONIA)) (MOTHER Y Y)) 2 + 12)
(20 NIL ((FATHER Y (HARMONIA)) (FATHER Y Y)) 3 + 12)
(21 NIL ((PARENT (ARES) (ARES))) 6 + 12)
(22 NIL ((MOTHER (ARES) (HARMONIA))) 2 + 14)
```

```
(23 NIL ((FATHER (ARES) (HARMONIA))) 3 + 14)
(24 NIL ((MOTHER (HARMONIA) (HARMONIA))) 2 + 18)
(25 NIL ((FATHER (HARMONIA) (HARMONIA))) 3 + 18)
(26 NIL ((MOTHER (ARES) (ARES))) 6 + 19)
(27 NIL ((FATHER (ARES) (ARES))) 6 + 20)
(6 + 23 = FALSE)
Theorem was successfully proven!

CG-USER(132): (unit-preference *HARMONIA* 7)

(1 ((GRANDPARENT X Y)) ((PARENT X Z) (PARENT Z Y)))
(2 ((PARENT X Y)) ((MOTHER X Y)))
(3 ((PARENT X Y)) ((FATHER X Y)))
(4 ((FATHER (ZEUS) (ARES))) NIL)
(5 ((MOTHER (HERA) (ARES))) NIL)
(6 ((FATHER (ARES) (HARMONIA))) NIL)
(7 NIL ((GRANDPARENT X (HARMONIA))))
(8 NIL ((PARENT X Z) (PARENT Z (HARMONIA))) 1 + 7)
(9 NIL ((PARENT Y (HARMONIA)) (MOTHER X Y)) 2 + 8)
(10 NIL ((PARENT X X) (MOTHER X (HARMONIA))) 2 + 8)
(11 NIL ((PARENT Y (HARMONIA)) (FATHER X Y)) 3 + 8)
(12 NIL ((PARENT X X) (FATHER X (HARMONIA))) 3 + 8)
(13 NIL ((MOTHER (HARMONIA) (HARMONIA)) (MOTHER (HARMONIA) (HARMONIA))) 2 + 9)
(14 NIL ((PARENT (ARES) (HARMONIA))) 5 + 9)
(15 NIL ((MOTHER (ARES) (HARMONIA))) 2 + 14)
(16 NIL ((FATHER (ARES) (HARMONIA))) 3 + 14)
(6 + 16 = FALSE)
Theorem was successfully proven!
```

(5)Harmonia (with Answer(x)):

```
CG-USER(134): (defvar *HarmoniaAnswer* '((1 ((Grandparent x y)) ((Parent x z) (Parent z y)))
                    (2 ((Parent x y)) ((Mother x y))) (3 ((Parent x y)) ((Father x y))) (4 ((Father (Zeus)
(Ares))) nil)
                    (5 ((Mother (Hera) (Ares))) nil) (6 ((Father (Ares) (Harmonia))) nil)
                    (7 ((Answer x)) ((Grandparent x (Harmonia))))))

CG-USER(135): (two-pointer *HarmoniaAnswer* 7)

(1 ((GRANDPARENT X Y)) ((PARENT X Z) (PARENT Z Y)))
(2 ((PARENT X Y)) ((MOTHER X Y)))
(3 ((PARENT X Y)) ((FATHER X Y)))
(4 ((FATHER (ZEUS) (ARES))) NIL)
(5 ((MOTHER (HERA) (ARES))) NIL)
(6 ((FATHER (ARES) (HARMONIA))) NIL)
```

```
(7 ((ANSWER X)) ((GRANDPARENT X (HARMONIA))))
(8 ((ANSWER X)) ((PARENT X Z) (PARENT Z (HARMONIA))) 1 + 7)
(9 ((ANSWER X)) ((PARENT Y (HARMONIA)) (MOTHER X Y)) 2 + 8)
(10 ((ANSWER X)) ((PARENT X X) (MOTHER X (HARMONIA))) 2 + 8)
(11 ((ANSWER X)) ((PARENT Y (HARMONIA)) (FATHER X Y)) 3 + 8)
(12 ((ANSWER X)) ((PARENT X X) (FATHER X (HARMONIA))) 3 + 8)
(13 ((ANSWER (HARMONIA))) ((MOTHER (HARMONIA) (HARMONIA)) (MOTHER (HARMONIA)
(HARMONIA))) 2 + 9)
(14 ((ANSWER (HERA))) ((PARENT (ARES) (HARMONIA))) 5 + 9)
(15 ((ANSWER Y)) ((MOTHER Y (HARMONIA)) (MOTHER Y Y)) 2 + 10)
(16 ((ANSWER Y)) ((MOTHER Y (HARMONIA)) (FATHER Y Y)) 3 + 10)
(17 ((ANSWER (HARMONIA))) ((FATHER (HARMONIA) (HARMONIA)) (FATHER (HARMONIA)
(HARMONIA))) 3 + 11)
(18 ((ANSWER (ZEUS))) ((PARENT (ARES) (HARMONIA))) 4 + 11)
(19 ((ANSWER (ARES))) ((PARENT (HARMONIA) (HARMONIA))) 6 + 11)
(20 ((ANSWER Y)) ((FATHER Y (HARMONIA)) (MOTHER Y Y)) 2 + 12)
(21 ((ANSWER Y)) ((FATHER Y (HARMONIA)) (FATHER Y Y)) 3 + 12)
(22 ((ANSWER (ARES))) ((PARENT (ARES) (ARES))) 6 + 12)
(23 ((ANSWER (HERA))) ((MOTHER (ARES) (HARMONIA))) 2 + 14)
(24 ((ANSWER (HERA))) ((FATHER (ARES) (HARMONIA))) 3 + 14)
(25 ((ANSWER (ZEUS))) ((MOTHER (ARES) (HARMONIA))) 2 + 18)
(26 ((ANSWER (ZEUS))) ((FATHER (ARES) (HARMONIA))) 3 + 18)
(27 ((ANSWER (ARES))) ((MOTHER (HARMONIA) (HARMONIA))) 2 + 19)
(28 ((ANSWER (ARES))) ((FATHER (HARMONIA) (HARMONIA))) 3 + 19)
(29 ((ANSWER (ARES))) ((MOTHER (ARES) (ARES))) 6 + 20)
(30 ((ANSWER (ARES))) ((FATHER (ARES) (ARES))) 6 + 21)
(ANSWER (HERA))
Answer was found!

CG-USER(136): (unit-preference *HarmoniaAnswer* 7)

(1 ((GRANDPARENT X Y)) ((PARENT X Z) (PARENT Z Y)))
(2 ((PARENT X Y)) ((MOTHER X Y)))
(3 ((PARENT X Y)) ((FATHER X Y)))
(4 ((FATHER (ZEUS) (ARES))) NIL)
(5 ((MOTHER (HERA) (ARES))) NIL)
(6 ((FATHER (ARES) (HARMONIA))) NIL)
(7 ((ANSWER X)) ((GRANDPARENT X (HARMONIA))))
(8 ((ANSWER X)) ((PARENT X Z) (PARENT Z (HARMONIA))) 1 + 7)
(9 ((ANSWER X)) ((PARENT Y (HARMONIA)) (MOTHER X Y)) 2 + 8)
(10 ((ANSWER X)) ((PARENT X X) (MOTHER X (HARMONIA))) 2 + 8)
(11 ((ANSWER X)) ((PARENT Y (HARMONIA)) (FATHER X Y)) 3 + 8)
(12 ((ANSWER X)) ((PARENT X X) (FATHER X (HARMONIA))) 3 + 8)
(13 ((ANSWER (HARMONIA))) ((MOTHER (HARMONIA) (HARMONIA)) (MOTHER (HARMONIA)
(HARMONIA))) 2 + 9)
(14 ((ANSWER (HERA))) ((PARENT (ARES) (HARMONIA))) 5 + 9)
(15 ((ANSWER (HERA))) ((MOTHER (ARES) (HARMONIA))) 2 + 14)
(16 ((ANSWER (HERA))) ((FATHER (ARES) (HARMONIA))) 3 + 14)
(ANSWER (HERA))
Answer was found!
```

(6) My theorem (without Answer(x))

```
(defvar *MyTheorem* '((1 ((VicePresident x)) ((OnTicket x y) (President y)))
                                          (2 ((President y)) ((WinElection y)))
                                          (3 ((WinElection (Trump))) nil)
                                          (4 ((OnTicket (Pence) (Trump))) nil)
                                          (5 nil ((VicePresident x)))))
```

```
CG-USER(2): (two-pointer *MYTHEOREM* 5)

(1 ((VICEPRESIDENT X)) ((ONTICKET X Y) (PRESIDENT Y)))
(2 ((PRESIDENT Y)) ((WINELECTION Y)))
(3 ((WINELECTION (TRUMP))) NIL)
(4 ((ONTICKET (PENCE) (TRUMP))) NIL)
(5 NIL ((VICEPRESIDENT X)))
(6 NIL ((ONTICKET X Y) (PRESIDENT Y)) 1 + 5)
(7 NIL ((ONTICKET X Y) (WINELECTION Y)) 2 + 6)
(8 NIL ((PRESIDENT (TRUMP))) 4 + 6)
(9 NIL ((ONTICKET X (TRUMP))) 3 + 7)
(10 NIL ((WINELECTION (TRUMP))) 4 + 7)
(4 + 9 = FALSE)
Theorem was successfully proven!
```

```
CG-USER(4): (unit-preference *MYTHEOREM* 5)

(1 ((VICEPRESIDENT X)) ((ONTICKET X Y) (PRESIDENT Y)))
(2 ((PRESIDENT Y)) ((WINELECTION Y)))
(3 ((WINELECTION (TRUMP))) NIL)
(4 ((ONTICKET (PENCE) (TRUMP))) NIL)
(5 NIL ((VICEPRESIDENT X)))
(6 NIL ((ONTICKET X Y) (PRESIDENT Y)) 1 + 5)
(7 NIL ((ONTICKET X Y) (WINELECTION Y)) 2 + 6)
(8 NIL ((PRESIDENT (TRUMP))) 4 + 6)
(9 NIL ((WINELECTION (TRUMP))) 2 + 8)
(3 + 9 = FALSE)
Theorem was successfully proven!
```

(7)My theorem (with Answer(x))

```
(defvar *MyTheo* '((1 ((VicePresident x)) ((OnTicket x y) (President y)))
                                           (2 ((President y)) ((WinElection y)))
                                           (3 ((WinElection (Trump))) nil)
                                           (4 ((Onticket (Pence) (Trump))) nil)
                                           (5 ((Answer x)) ((VicePresident x)))))
```

```
CG-USER(8): (two-pointer *MYTHEO* 5)

(1 ((VICEPRESIDENT X)) ((ONTICKET X Y) (PRESIDENT Y)))
(2 ((PRESIDENT Y)) ((WINELECTION Y)))
(3 ((WINELECTION (TRUMP))) NIL)
(4 ((ONTICKET (PENCE) (TRUMP))) NIL)
(5 ((ANSWER X)) ((VICEPRESIDENT X)))
(6 ((ANSWER X)) ((ONTICKET X Y) (PRESIDENT Y)) 1 + 5)
(7 ((ANSWER X)) ((ONTICKET X Y) (WINELECTION Y)) 2 + 6)
(8 ((ANSWER (PENCE))) ((PRESIDENT (TRUMP))) 4 + 6)
(9 ((ANSWER X)) ((ONTICKET X (TRUMP))) 3 + 7)
(10 ((ANSWER (PENCE))) ((WINELECTION (TRUMP))) 4 + 7)
(ANSWER (PENCE))
Answer was found!
```

```
CG-USER(9): (unit-preference *MYTHEO* 5)

(1 ((VICEPRESIDENT X)) ((ONTICKET X Y) (PRESIDENT Y)))
(2 ((PRESIDENT Y)) ((WINELECTION Y)))
(3 ((WINELECTION (TRUMP))) NIL)
(4 ((ONTICKET (PENCE) (TRUMP))) NIL)
(5 ((ANSWER X)) ((VICEPRESIDENT X)))
(6 ((ANSWER X)) ((ONTICKET X Y) (PRESIDENT Y)) 1 + 5)
(7 ((ANSWER X)) ((ONTICKET X Y) (WINELECTION Y)) 2 + 6)
(8 ((ANSWER (PENCE))) ((PRESIDENT (TRUMP))) 4 + 6)
(9 ((ANSWER (PENCE))) ((WINELECTION (TRUMP))) 2 + 8)
(ANSWER (PENCE))
Answer was found!
```