Handbook of Practical Logic and Automated Reasoning

OCaml run results

Following are the results of running the code from the book using OCaml.

The order for running the code is taken from this web page.

http://www.cl.cam.ac.uk/~jrh13/atp/

The results have been enhanced with color coding and traceability comments.

Note: This is not a final version, as all of the possible examples from the book are not in this version.

Lines in green are typically the output of a function, but sometimes are other details such a value. These are used as evidence of the correctness of the conversion to F#.

```
- : expression = Add (Mul (Const 2, Var "x"), Var "y")
```

Comments in blue are traceability comments for tracing an example between the OCaml code, OCaml output, F# example, and F# unit test. Traceability here refers to being able to cross reference an example among all of the locations; the word traceability is used because you typically want to trace a problem in a test back to the original OCaml code. Since this is a line of succession, it is more than a cross reference. These comments were added by hand.

```
// intro.p001
```

Lines in red show warnings. The corresponding F# warning (FS0025) has been disabled in the F# output.

```
File "/home/student/Harrison/OCaml/prop.ml", line 54, characters 2-263:
```

lib.ml

```
#use "lib.ml";;
```

```
# #use "lib.ml";;
val identity : 'a -> 'a = <fun>
val ( ** ) : ('a -> 'b) -> ('c -> 'a) -> 'c -> 'b = <fun>
val gcd_num : num -> num -> num = <fun>
val lcm_num : num -> num -> num = <fun>
val non : ('a -> bool) -> 'a -> bool = <fun>
val check : ('a -> bool) -> 'a -> 'a = <fun>
val funpow : int -> ('a -> 'a) -> 'a -> 'a = <fun>
val can : ('a -> 'b) -> 'a -> bool = <fun>
val repeat : ('a -> 'a) -> 'a -> 'a = <fun>
val ( -- ) : int -> int -> int list = <fun>
val ( --- ) : num -> num -> num list = <fun>
val map2 : ('a -> 'b -> 'c) -> 'a list -> 'b list -> 'c list = <fun>
val rev : 'a list -> 'a list = <fun>
val hd : 'a list -> 'a = <fun>
val tl : 'a list -> 'a list = <fun>
val itlist : ('a -> 'b -> 'b) -> 'a list -> 'b -> 'b = <fun>
val end_itlist : ('a -> 'a -> 'a) -> 'a list -> 'a = <fun>
val itlist2 : ('a -> 'b -> 'c -> 'c) -> 'a list -> 'b list -> 'c -> 'c =
  <fun>
val zip : 'a list -> 'b list -> ('a * 'b) list = <fun>
val forall : ('a -> bool) -> 'a list -> bool = <fun>
val exists : ('a -> bool) -> 'a list -> bool = <fun>
val partition : ('a -> bool) -> 'a list -> 'a list * 'a list = <fun>
val filter : ('a -> bool) -> 'a list -> 'a list = <fun>
val length : 'a list -> int = <fun>
val last : 'a list -> 'a = <fun>
val butlast : 'a list -> 'a list = <fun>
val find : ('a -> bool) -> 'a list -> 'a = <fun>
val el : int -> 'a list -> 'a = <fun>
val map : ('a -> 'b) -> 'a list -> 'b list = <fun>
val allpairs : ('a -> 'b -> 'c) -> 'a list -> 'b list -> 'c list =
  <fun>
val distinctpairs : 'a list -> ('a * 'a) list = <fun>
val chop_list : int -> 'a list -> 'a list * 'a list = <fun>
val replicate : int -> 'a -> 'a list = <fun>
val insertat : int -> 'a -> 'a list -> 'a list = <fun>
val forall2 : ('a -> 'b -> bool) -> 'a list -> 'b list -> bool = <fun>
val index : 'a -> 'a list -> int = <fun>
val unzip : ('a * 'b) list -> 'a list * 'b list = <fun>
val earlier : 'a list -> 'a -> 'a -> bool = <fun>
val do_list : ('a -> 'b) -> 'a list -> unit = <fun>
val assoc : 'a -> ('a * 'b) list -> 'b = <fun>
val rev_assoc : 'a -> ('b * 'a) list -> 'b = <fun>
val merge : ('a -> 'a -> bool) -> 'a list -> 'a list -> 'a list = <fun>
val sort : ('a -> 'a -> bool) -> 'a list -> 'a list = <fun>
val increasing : ('a -> 'b) -> 'a -> 'a -> bool = <fun>
val decreasing : ('a -> 'b) -> 'a -> 'a -> bool = <fun>
val uniq : 'a list -> 'a list = <fun>
val repetitions : 'a list -> ('a * int) list = <fun>
val tryfind : ('a -> 'b) -> 'a list -> 'b = <fun>
val mapfilter : ('a -> 'b) -> 'a list -> 'b list = <fun>
val optimize : ('a -> 'a -> bool) -> ('b -> 'a) -> 'b list -> 'b =
  <fun>
val maximize : ('a -> 'b) -> 'a list -> 'a = <fun>
val minimize : ('a -> 'b) -> 'a list -> 'a = <fun>
val setify : 'a list -> 'a list = <fun>
val union : 'a list -> 'a list -> 'a list = <fun>
val intersect : 'a list -> 'a list -> 'a list = <fun>
val subtract : 'a list -> 'a list -> 'a list = <fun>
val subset : 'a list -> 'a list -> bool = <fun>
val psubset : 'a list -> 'a list -> bool = <fun>
val set_eq : 'a list -> 'a list -> bool = <fun>
val insert : 'a -> 'a list -> 'a list = <fun>
val image : ('a -> 'b) -> 'a list -> 'b list = <fun>
val unions : 'a list list -> 'a list = <fun>
val mem : 'a -> 'a list -> bool = <fun>
val allsets : int -> 'a list -> 'a list list = <fun>
val allsubsets : 'a list -> 'a list list = <fun>
val allnonemptysubsets : 'a list -> 'a list list = <fun>
val explode : string -> string list = <fun>
val implode : string list -> string = <fun>
val time : ('a -> 'b) -> 'a -> 'b = <fun>
type ('a, 'b) func =
    Empty
  | Leaf of int * ('a * 'b) list
  | Branch of int * int * ('a, 'b) func * ('a, 'b) func
val undefined : ('a, 'b) func = Empty
val is_undefined : ('a, 'b) func -> bool = <fun>
val mapf : ('a -> 'b) -> ('c, 'a) func -> ('c, 'b) func = <fun>
val foldl : ('a -> 'b -> 'c -> 'a) -> 'a -> ('b, 'c) func -> 'a = <fun>
val foldr : ('a -> 'b -> 'c -> 'c) -> ('a, 'b) func -> 'c -> 'c = <fun>
val graph : ('a, 'b) func -> ('a * 'b) list = <fun>
val dom : ('a, 'b) func -> 'a list = <fun>
val ran : ('a, 'b) func -> 'b list = <fun>
val applyd : ('a, 'b) func -> ('a -> 'b) -> 'a -> 'b = <fun>
val apply : ('a, 'b) func -> 'a -> 'b = <fun>
val tryapplyd : ('a, 'b) func -> 'a -> 'b -> 'b = <fun>
val tryapplyl : ('a, 'b list) func -> 'a -> 'b list = <fun>
val defined : ('a, 'b) func -> 'a -> bool = <fun>
val undefine : 'a -> ('a, 'b) func -> ('a, 'b) func = <fun>
val ( |-> ) : 'a -> 'b -> ('a, 'b) func -> ('a, 'b) func = <fun>
val combine :
  ('a -> 'a -> 'a) ->
  ('a -> bool) -> ('b, 'a) func -> ('b, 'a) func -> ('b, 'a) func =
  <fun>
val ( |=> ) : 'a -> 'b -> ('a, 'b) func = <fun>
val fpf : 'a list -> 'b list -> ('a, 'b) func = <fun>
val choose : ('a, 'b) func -> 'a * 'b = <fun>
val print_fpf : ('a, 'b) func -> unit = <fun>
val valmod : 'a -> 'b -> ('a -> 'b) -> 'a -> 'b = <fun>
val undef : 'a -> 'b = <fun>
type 'a pnode = Nonterminal of 'a | Terminal of 'a * int
type 'a partition = Partition of ('a, 'a pnode) func
```

```
val terminus : 'a partition -> 'a -> 'a * int = <fun>
val tryterminus : 'a partition -> 'a -> 'a * int = <fun>
val canonize : 'a partition -> 'a -> 'a = <fun>
val equivalent : 'a partition -> 'a -> 'a -> bool = <fun>
val equate : 'a * 'a -> 'a partition -> 'a partition = <fun>
val unequal : 'a partition = Partition <func>
val equated : 'a partition -> 'a list = <fun>
val first : num -> (num -> bool) -> num = <fun>
#
```

intro.ml

```
#use "intro.ml";;
```

```
# #use "intro.ml";;
type expression =
    Var of string
  | Const of int
  | Add of expression * expression
  | Mul of expression * expression
- : dummy_interactive = START_INTERACTIVE
// intro.p001
- : expression = Add (Mul (Const 2, Var "x"), Var "y")
- : dummy_interactive = END_INTERACTIVE
val simplify1 : expression -> expression = <fun>
val simplify : expression -> expression = <fun>
- : dummy_interactive = START_INTERACTIVE
// intro.p002
val e : expression =
  Add (Mul (Add (Mul (Const 0, Var "x"), Const 1), Const 3), Const 12)
- : expression = Const 15
- : dummy_interactive = END_INTERACTIVE
val matches : string -> string -> bool = <fun>
val space : string -> bool = <fun>
val punctuation : string -> bool = <fun>
val symbolic : string -> bool = <fun>
val numeric : string -> bool = <fun>
val alphanumeric : string -> bool = <fun>
val lexwhile : (string -> bool) -> string list -> string * string list =
  <fun>
val lex : string list -> string list = <fun>
- : dummy_interactive = START_INTERACTIVE
// intro.p003
- : string list =
["2"; "*"; "("; "("; "var_1"; "+"; "x'"; ")"; "+"; "11"; ")"]
// intro.p004
- : string list =
["if"; "("; "*"; "p1"; "--"; "=="; "*"; "p2"; "++"; ")"; "then"; "f";
 "("; ")"; "else"; "g"; "("; ")"]
- : dummy_interactive = END_INTERACTIVE
val parse_expression : string list -> expression * string list = <fun>
val parse_product : string list -> expression * string list = <fun>
val parse_atom : string list -> expression * string list = <fun>
val make_parser : (string list -> 'a * 'b list) -> string -> 'a = <fun>
val default_parser : string -> expression = <fun>
- : dummy_interactive = START_INTERACTIVE
// intro.p005
- : expression = Add (Var "x", Const 1)
// intro.p006
- : expression =
Mul (Add (Var "x1", Add (Var "x2", Var "x3")),
 Add (Const 1, Add (Const 2, Add (Mul (Const 3, Var "x"), Var "y"))))
- : dummy_interactive = END_INTERACTIVE
```

```
val string_of_exp : expression -> string = <fun>
- : dummy_interactive = START_INTERACTIVE
// intro.p007
- : string = "(x + (3 * y))"
- : dummy_interactive = END_INTERACTIVE
val string_of_exp : int -> expression -> string = <fun>
val print_exp : expression -> unit = <fun>
- : dummy_interactive = START_INTERACTIVE
// intro.p008
- : expression = <<x + 3 * y>>
// intro.p009
- : expression = <<(x + 3) * y>>
// intro.p010
- : expression = <<1 + 2 + 3>>
// intro.p011
- : expression = <<((1 + 2) + 3) + 4>>
- : dummy_interactive = END_INTERACTIVE
- : dummy_interactive = START_INTERACTIVE
// intro.p012
- : expression =
<<(x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10) * (y1 + y2 + y3 + y4 + y5 + y6 + y7
+ y8 + y9 + y10)>>
- : dummy_interactive = END_INTERACTIVE
#
```

formulas.ml

```
#use "formulas.ml";;
```

```
# #use "formulas.ml";;
type 'a formula =
    False
  | True
  | Atom of 'a
  | Not of 'a formula
  | And of 'a formula * 'a formula
  | Or of 'a formula * 'a formula
  | Imp of 'a formula * 'a formula
  | Iff of 'a formula * 'a formula
  | Forall of string * 'a formula
  | Exists of string * 'a formula
val parse_ginfix :
  'a ->
  (('b -> 'c) -> 'b -> 'b -> 'c) ->
  ('b -> 'c) -> ('a list -> 'b * 'a list) -> 'a list -> 'c * 'a list =
  <fun>
val parse_left_infix :
  'a ->
  ('b * 'b -> 'b) ->
  ('a list -> 'b * 'a list) -> 'a list -> 'b * 'a list = <fun>
val parse_right_infix :
  'a ->
  ('b * 'b -> 'b) ->
  ('a list -> 'b * 'a list) -> 'a list -> 'b * 'a list = <fun>
val parse_list :
  'a -> ('a list -> 'b * 'a list) -> 'a list -> 'b list * 'a list =
  <fun>
val papply : ('a -> 'b) -> 'a * 'c -> 'b * 'c = <fun>
val nextin : 'a list -> 'a -> bool = <fun>
val parse_bracketed : ('a -> 'b * 'c list) -> 'c -> 'a -> 'b * 'c list =
  <fun>
val parse_atomic_formula :
  (string list -> string list -> 'a formula * string list) *
  (string list -> string list -> 'a formula * string list) ->
  string list -> string list -> 'a formula * string list = <fun>
val parse_quant :
  (string list -> string list -> 'a formula * string list) *
  (string list -> string list -> 'a formula * string list) ->
  string list ->
  (string * 'a formula -> 'a formula) ->
  string -> string list -> 'a formula * string list = <fun>
val parse_formula :
  (string list -> string list -> 'a formula * string list) *
  (string list -> string list -> 'a formula * string list) ->
  string list -> string list -> 'a formula * string list = <fun>
val bracket : bool -> int -> ('a -> 'b -> 'c) -> 'a -> 'b -> unit =
  <fun>
val strip_quant : 'a formula -> string list * 'a formula = <fun>
val print_formula : (int -> 'a -> unit) -> 'a formula -> unit = <fun>
val print_qformula : (int -> 'a -> unit) -> 'a formula -> unit = <fun>
val mk_and : 'a formula -> 'a formula -> 'a formula = <fun>
val mk_or : 'a formula -> 'a formula -> 'a formula = <fun>
val mk_imp : 'a formula -> 'a formula -> 'a formula = <fun>
val mk_iff : 'a formula -> 'a formula -> 'a formula = <fun>
val mk_forall : string -> 'a formula -> 'a formula = <fun>
val mk_exists : string -> 'a formula -> 'a formula = <fun>
val dest_iff : 'a formula -> 'a formula * 'a formula = <fun>
val dest_and : 'a formula -> 'a formula * 'a formula = <fun>
val conjuncts : 'a formula -> 'a formula list = <fun>
val dest_or : 'a formula -> 'a formula * 'a formula = <fun>
val disjuncts : 'a formula -> 'a formula list = <fun>
val dest_imp : 'a formula -> 'a formula * 'a formula = <fun>
val antecedent : 'a formula -> 'a formula = <fun>
val consequent : 'a formula -> 'a formula = <fun>
val onatoms : ('a -> 'a formula) -> 'a formula -> 'a formula = <fun>
val overatoms : ('a -> 'b -> 'b) -> 'a formula -> 'b -> 'b = <fun>
val atom_union : ('a -> 'b list) -> 'a formula -> 'b list = <fun>
#
```

prop.ml

```
#use "prop.ml";;
```

```
#use "prop.ml";;
type prop = P of string
val pname : prop -> string = <fun>
val parse_propvar : 'a -> string list -> prop formula * string list =
  <fun>
val parse_prop_formula : string -> prop formula = <fun>
val default_parser : string -> prop formula = <fun>
val print_propvar : 'a -> prop -> unit = <fun>
val print_prop_formula : prop formula -> unit = <fun>
- : dummy_interactive = START_INTERACTIVE
val fm : prop formula = <<p ==> q <=> r /\ s \/ (t <=> ~(~u) /\ v)>>
// prop.p001
- : prop formula =
<<(p ==> q <=> r /\ s \/ (t <=> ~(~u) /\ v)) /\
  (p ==> q <=> r /\ s \/ (t <=> ~(~u) /\ v))>>
// prop.p002
- : prop formula =
<<((p ==> q <=> r /\ s \/ (t <=> ~(~u) /\ v)) \/
   (p ==> q <=> r /\ s \/ (t <=> ~(~u) /\ v))) /\
  (p ==> q <=> r /\ s \/ (t <=> ~(~u) /\ v))>>
- : dummy_interactive = END_INTERACTIVE
File "/home/student/Harrison/OCaml/prop.ml", line 54, characters 2-263:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
(Exists (_, _)|Forall (_, _))
val eval : 'a formula -> ('a -> bool) -> bool = <fun>
- : dummy_interactive = START_INTERACTIVE
File "/home/student/Harrison/OCaml/prop.ml", line 70, characters 6-58:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
P ""
// prop.p007
- : bool = true
File "/home/student/Harrison/OCaml/prop.ml", line 73, characters 6-58:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
P ""
// prop.p008
- : bool = false
- : dummy_interactive = END_INTERACTIVE
val atoms : 'a formula -> 'a list = <fun>
- : dummy_interactive = START_INTERACTIVE
// prop.p009
- : prop list = [P "p"; P "q"; P "r"; P "s"]
- : dummy_interactive = END_INTERACTIVE
val onallvaluations :
  (('a -> bool) -> bool) -> ('a -> bool) -> 'a list -> bool = <fun>
val print_truthtable : prop formula -> unit = <fun>
- : dummy_interactive = START_INTERACTIVE
```

```
// prop.p010
p     q     r     | formula
---------------------------
false false false | true
false false true  | true
false true  false | true
false true  true  | true
true  false false | true
true  false true  | true
true  true  false | false
true  true  true  | true
---------------------------
- : unit = ()
// prop.p011
val fm : prop formula = <<p /\ q ==> q /\ r>>
p     q     r     | formula
---------------------------
false false false | true
false false true  | true
false true  false | true
false true  true  | true
true  false false | true
true  false true  | true
true  true  false | false
true  true  true  | true
---------------------------
- : unit = ()
- : dummy_interactive = END_INTERACTIVE
- : dummy_interactive = START_INTERACTIVE
// prop.p012
p     q     | formula
---------------------
false false | true
false true  | true
true  false | true
true  true  | true
---------------------
- : unit = ()
// prop.p013
p     | formula
---------------
false | false
true  | false
---------------
- : unit = ()
- : dummy_interactive = END_INTERACTIVE
val tautology : 'a formula -> bool = <fun>
- : dummy_interactive = START_INTERACTIVE
// prop.p014
- : bool = true
// prop.p015
- : bool = false
// prop.p016
- : bool = false
// prop.p017
```

```
- : bool = true
- : dummy_interactive = END_INTERACTIVE
val unsatisfiable : 'a formula -> bool = <fun>
val satisfiable : 'a formula -> bool = <fun>
val psubst : ('a, 'a formula) func -> 'a formula -> 'a formula = <fun>
- : dummy_interactive = START_INTERACTIVE
// prop.p018
- : prop formula = <<(p /\ q) /\ q /\ (p /\ q) /\ q>>
- : dummy_interactive = END_INTERACTIVE
- : dummy_interactive = START_INTERACTIVE
// prop.p019
- : bool = true
// prop.p020
- : bool = true
// prop.p021
- : bool = true
// prop.p022
- : bool = true
// prop.p023
- : bool = true
// prop.p024
- : bool = false
// prop.p025
- : bool = true
- : dummy_interactive = END_INTERACTIVE
val dual : 'a formula -> 'a formula = <fun>
- : dummy_interactive = START_INTERACTIVE
// prop.p026
- : prop formula = <<p /\ ~p>>
- : dummy_interactive = END_INTERACTIVE
val psimplify1 : 'a formula -> 'a formula = <fun>
val psimplify : 'a formula -> 'a formula = <fun>
- : dummy_interactive = START_INTERACTIVE
// prop.p027
- : prop formula = <<~x ==> ~y>>
// prop.p028
- : prop formula = <<true>>
- : dummy_interactive = END_INTERACTIVE
val negative : 'a formula -> bool = <fun>
val positive : 'a formula -> bool = <fun>
val negate : 'a formula -> 'a formula = <fun>
val nnf : 'a formula -> 'a formula = <fun>
val nnf : 'a formula -> 'a formula = <fun>
- : dummy_interactive = START_INTERACTIVE
val fm : prop formula = <<(p <=> q) <=> ~(r ==> s)>>
val fm' : prop formula =
  <<(p /\ q \/ ~p /\ ~q) /\ r /\ ~s \/
    (p /\ ~q \/ ~p /\ q) /\ (~r \/ s)>>
// prop.p029
- : bool = true
- : dummy_interactive = END_INTERACTIVE
val nenf : 'a formula -> 'a formula = <fun>
val nenf : 'a formula -> 'a formula = <fun>
- : dummy_interactive = START_INTERACTIVE
// prop.p030
```

```
- : bool = true
// prop.p031
- : bool = true
- : dummy_interactive = END_INTERACTIVE
val list_conj : 'a formula list -> 'a formula = <fun>
val list_disj : 'a formula list -> 'a formula = <fun>
val mk_lits : 'a formula list -> ('a -> bool) -> 'a formula = <fun>
val allsatvaluations :
  (('a -> bool) -> bool) ->
  ('a -> bool) -> 'a list -> ('a -> bool) list = <fun>
val dnf : 'a formula -> 'a formula = <fun>
- : dummy_interactive = START_INTERACTIVE
val fm : prop formula = <<(p \/ q /\ r) /\ (~p \/ ~r)>>
// prop.p032
- : prop formula = <<~p /\ q /\ r \/ p /\ ~q /\ ~r \/ p /\ q /\ ~r>>
// prop.p033
p     q     r     | formula
---------------------------
false false false | false
false false true  | false
false true  false | false
false true  true  | true
true  false false | true
true  false true  | false
true  true  false | true
true  true  true  | false
---------------------------
- : unit = ()
// prop.p034
- : prop formula =
<<~p /\ ~q /\ ~r /\ ~s /\ ~t /\ u /\ v \/
  ~p /\ ~q /\ ~r /\ ~s /\ t /\ u /\ v \/
  ~p /\ ~q /\ ~r /\ s /\ ~t /\ u /\ v \/
  ~p /\ ~q /\ ~r /\ s /\ t /\ u /\ v \/
  ~p /\ ~q /\ r /\ ~s /\ ~t /\ u /\ v \/
  ~p /\ ~q /\ r /\ ~s /\ t /\ u /\ v \/
  ~p /\ ~q /\ r /\ s /\ ~t /\ u /\ v \/
  ~p /\ ~q /\ r /\ s /\ t /\ u /\ v \/
  ~p /\ q /\ ~r /\ ~s /\ ~t /\ u /\ v \/
  ~p /\ q /\ ~r /\ ~s /\ t /\ u /\ v \/
  ~p /\ q /\ ~r /\ s /\ ~t /\ u /\ v \/
  ~p /\ q /\ ~r /\ s /\ t /\ u /\ v \/
  ~p /\ q /\ r /\ ~s /\ ~t /\ u /\ v \/
  ~p /\ q /\ r /\ ~s /\ t /\ u /\ v \/
  ~p /\ q /\ r /\ s /\ ~t /\ u /\ v \/
  ~p /\ q /\ r /\ s /\ t /\ u /\ v \/
  p /\ ~q /\ ~r /\ ~s /\ ~t /\ u /\ v \/
  p /\ ~q /\ ~r /\ ~s /\ t /\ u /\ v \/
  p /\ ~q /\ ~r /\ s /\ ~t /\ u /\ v \/
  p /\ ~q /\ ~r /\ s /\ t /\ u /\ v \/
  p /\ ~q /\ r /\ ~s /\ ~t /\ u /\ v \/
  p /\ ~q /\ r /\ ~s /\ t /\ u /\ v \/
  p /\ ~q /\ r /\ s /\ ~t /\ u /\ v \/
  p /\ ~q /\ r /\ s /\ t /\ u /\ v \/
  p /\ q /\ ~r /\ ~s /\ ~t /\ u /\ v \/
  p /\ q /\ ~r /\ ~s /\ t /\ u /\ v \/
  p /\ q /\ ~r /\ s /\ ~t /\ u /\ v \/
  p /\ q /\ ~r /\ s /\ t /\ u /\ v \/
  p /\ q /\ r /\ ~s /\ ~t /\ u /\ v \/
  p /\ q /\ r /\ ~s /\ t /\ u /\ v \/
  p /\ q /\ r /\ s /\ ~t /\ u /\ v \/
  p /\ q /\ r /\ s /\ t /\ u /\ ~v \/ p /\ q /\ r /\ s /\ t /\ u /\ v>>
- : dummy_interactive = END_INTERACTIVE
val distrib : 'a formula -> 'a formula = <fun>
val rawdnf : 'a formula -> 'a formula = <fun>
- : dummy_interactive = START_INTERACTIVE
// prop.p035
- : prop formula =
<<(p /\ ~p \/ (q /\ r) /\ ~p) \/ p /\ ~r \/ (q /\ r) /\ ~r>>
- : dummy_interactive = END_INTERACTIVE
val distrib : 'a list list -> 'a list list -> 'a list list = <fun>
val purednf : 'a formula -> 'a formula list list = <fun>
- : dummy_interactive = START_INTERACTIVE
// prop.p036
- : prop formula list list =
[[<<p>>; <<~p>>]; [<<p>>; <<~r>>]; [<<q>>; <<r>>; <<~p>>];
 [<<q>>; <<r>>; <<~r>>]]
- : dummy_interactive = END_INTERACTIVE
val trivial : 'a formula list -> bool = <fun>
- : dummy_interactive = START_INTERACTIVE
// prop.p037
- : prop formula list list = [[<<p>>; <<~r>>]; [<<q>>; <<r>>; <<~p>>]]
- : dummy_interactive = END_INTERACTIVE
val simpdnf : 'a formula -> 'a formula list list = <fun>
val dnf : 'a formula -> 'a formula = <fun>
- : dummy_interactive = START_INTERACTIVE
val fm : prop formula = <<(p \/ q /\ r) /\ (~p \/ ~r)>>
// prop.p038
- : prop formula = <<p /\ ~r \/ q /\ r /\ ~p>>
// prop.p039
- : bool = true
- : dummy_interactive = END_INTERACTIVE
val purecnf : 'a formula -> 'a formula list list = <fun>
val simpcnf : 'a formula -> 'a formula list list = <fun>
val cnf : 'a formula -> 'a formula = <fun>
- : dummy_interactive = START_INTERACTIVE
val fm : prop formula = <<(p \/ q /\ r) /\ (~p \/ ~r)>>
// prop.p040
- : prop formula = <<(p \/ q) /\ (p \/ r) /\ (~p \/ ~r)>>
// prop.p041
- : bool = true
- : dummy_interactive = END_INTERACTIVE
#
```

propexamples.ml

```
#use "propexamples.ml";;
```

```
# #use "propexamples.ml";;
File "/home/student/Harrison/OCaml/propexamples.ml", line 15, characters 7-68:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
[]
val ramsey : int -> int -> int -> prop formula = <fun>
- : dummy_interactive = START_INTERACTIVE
// propexamples.p001
- : prop formula =
<<(p_1_2 /\ p_1_3 /\ p_2_3 \/
   p_1_2 /\ p_1_4 /\ p_2_4 \/
   p_1_3 /\ p_1_4 /\ p_3_4 \/ p_2_3 /\ p_2_4 /\ p_3_4) \/
  ~p_1_2 /\ ~p_1_3 /\ ~p_2_3 \/
  ~p_1_2 /\ ~p_1_4 /\ ~p_2_4 \/
  ~p_1_3 /\ ~p_1_4 /\ ~p_3_4 \/ ~p_2_3 /\ ~p_2_4 /\ ~p_3_4>>
// propexamples.p002
- : bool = false
// propexamples.p003
- : bool = true
- : dummy_interactive = END_INTERACTIVE
val halfsum : 'a formula -> 'a formula -> 'a formula = <fun>
val halfcarry : 'a formula -> 'a formula -> 'a formula = <fun>
val ha :
  'a formula -> 'a formula -> 'a formula -> 'a formula -> 'a formula =
  <fun>
val carry : 'a formula -> 'a formula -> 'a formula -> 'a formula =
  <fun>
val sum : 'a formula -> 'a formula -> 'a formula -> 'a formula = <fun>
val fa :
  'a formula ->
  'a formula -> 'a formula -> 'a formula -> 'a formula -> 'a formula =
  <fun>
val conjoin : ('a -> 'b formula) -> 'a list -> 'b formula = <fun>
val ripplecarry :
  (int -> 'a formula) ->
  (int -> 'a formula) ->
  (int -> 'a formula) -> (int -> 'a formula) -> int -> 'a formula =
  <fun>
val mk_index : string -> int -> prop formula = <fun>
val mk_index2 : string -> int -> int -> prop formula = <fun>
- : dummy_interactive = START_INTERACTIVE
File "/home/student/Harrison/OCaml/propexamples.ml", line 76, characters 4-18:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
[]
val x : int -> prop formula = <fun>
val y : int -> prop formula = <fun>
val out : int -> prop formula = <fun>
val c : int -> prop formula = <fun>
```

```
// propexamples.p004
- : prop formula =
<<((OUT_0 <=> (X_0 <=> ~Y_0) <=> ~C_0) /\
   (C_1 <=> X_0 /\ Y_0 \/ (X_0 \/ Y_0) /\ C_0)) /\
  (OUT_1 <=> (X_1 <=> ~Y_1) <=> ~C_1) /\
  (C_2 <=> X_1 /\ Y_1 \/ (X_1 \/ Y_1) /\ C_1)>>
- : dummy_interactive = END_INTERACTIVE
val ripplecarry0 :
  (int -> 'a formula) ->
  (int -> 'a formula) ->
  (int -> 'a formula) -> (int -> 'a formula) -> int -> 'a formula =
  <fun>
val ripplecarry1 :
  (int -> 'a formula) ->
  (int -> 'a formula) ->
  (int -> 'a formula) -> (int -> 'a formula) -> int -> 'a formula =
  <fun>
val mux : 'a formula -> 'a formula -> 'a formula -> 'a formula = <fun>
val offset : int -> (int -> 'a) -> int -> 'a = <fun>
val carryselect :
  (int -> 'a formula) ->
  (int -> 'a formula) ->
  (int -> 'a formula) ->
  (int -> 'a formula) ->
  (int -> 'a formula) ->
  (int -> 'a formula) ->
  (int -> 'a formula) ->
  (int -> 'a formula) -> int -> int -> 'a formula = <fun>
File "/home/student/Harrison/OCaml/propexamples.ml", line 120, characters 6-42:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
[]
val mk_adder_test : int -> int -> prop formula = <fun>
val rippleshift :
  (int -> 'a formula) ->
  (int -> 'a formula) ->
  (int -> 'a formula) ->
  'a formula -> (int -> 'a formula) -> int -> 'a formula = <fun>
val multiplier :
  (int -> int -> 'a formula) ->
  (int -> int -> 'a formula) ->
  (int -> int -> 'a formula) ->
  (int -> 'a formula) -> int -> 'a formula = <fun>
val bitlength : int -> int = <fun>
val bit : int -> int -> bool = <fun>
val congruent_to : (int -> 'a formula) -> int -> int -> 'a formula =
  <fun>
File "/home/student/Harrison/OCaml/propexamples.ml", line 171, characters 6-17:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
[]
File "/home/student/Harrison/OCaml/propexamples.ml", line 173, characters 6-12:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
[]
```

```
val prime : int -> prop formula = <fun>
- : dummy_interactive = START_INTERACTIVE
// propexamples.p005
- : bool = true
// propexamples.p006
- : bool = false
// propexamples.p007
- : bool = true
- : dummy_interactive = END_INTERACTIVE
#
```

defcnf.ml

```
#use "defcnf.ml";;
```

```
# #use "defcnf.ml";;
- : dummy_interactive = START_INTERACTIVE
// defcnf.p001
- : prop formula =
<<(p \/ q \/ r) /\
  (p \/ ~q \/ ~r) /\ (q \/ ~p \/ ~r) /\ (r \/ ~p \/ ~q)>>
- : dummy_interactive = END_INTERACTIVE
val mkprop : num -> prop formula * num = <fun>
val maincnf :
  prop formula * (prop formula, prop formula * prop formula) func * num ->
  prop formula * (prop formula, prop formula * prop formula) func * num =
  <fun>
val defstep :
  (prop formula -> prop formula -> prop formula) ->
  prop formula * prop formula ->
  prop formula * (prop formula, prop formula * prop formula) func * num ->
  prop formula * (prop formula, prop formula * prop formula) func * num =
  <fun>
val max_varindex : string -> string -> num -> num = <fun>
val mk_defcnf :
  (prop formula * ('a, 'b) func * num ->
   'c formula * ('d, 'e * 'c formula) func * 'f) ->
  prop formula -> 'c formula list list = <fun>
val defcnf : prop formula -> prop formula = <fun>
- : dummy_interactive = START_INTERACTIVE
// defcnf.p002
- : prop formula =
<<(p \/ p_1 \/ ~p_2) /\
  (p_1 \/ r \/ ~q) /\
  (p_2 \/ ~p) /\
  (p_2 \/ ~p_1) /\
  (p_2 \/ ~p_3) /\
  p_3 /\
  (p_3 \/ ~p_2 \/ ~s) /\ (q \/ ~p_1) /\ (s \/ ~p_3) /\ (~p_1 \/ ~r)>>
- : dummy_interactive = END_INTERACTIVE
val subcnf :
  ('a * 'b * 'c -> 'd * 'b * 'c) ->
  ('d -> 'd -> 'e) -> 'a * 'a -> 'f * 'b * 'c -> 'e * 'b * 'c = <fun>
val orcnf :
  prop formula * (prop formula, prop formula * prop formula) func * num ->
  prop formula * (prop formula, prop formula * prop formula) func * num =
  <fun>
val andcnf :
  prop formula * (prop formula, prop formula * prop formula) func * num ->
  prop formula * (prop formula, prop formula * prop formula) func * num =
  <fun>
val defcnfs : prop formula -> prop formula list list = <fun>
val defcnf : prop formula -> prop formula = <fun>
- : dummy_interactive = START_INTERACTIVE
```

```
// defcnf.p003
- : prop formula =
<<(p \/ p_1) /\ (p_1 \/ r \/ ~q) /\ (q \/ ~p_1) /\ s /\ (~p_1 \/ ~r)>>
- : dummy_interactive = END_INTERACTIVE
val andcnf3 :
  prop formula * (prop formula, prop formula * prop formula) func * num ->
  prop formula * (prop formula, prop formula * prop formula) func * num =
  <fun>
val defcnf3 : prop formula -> prop formula = <fun>
#
```

dp.ml

```
#use "dp.ml";;
```

```
# #use "dp.ml";;
val one_literal_rule : 'a formula list list -> 'a formula list list =
  <fun>
val affirmative_negative_rule :
  'a formula list list -> 'a formula list list = <fun>
val resolve_on :
  'a formula -> 'a formula list list -> 'a formula list list = <fun>
val resolution_blowup : 'a formula list list -> 'a formula -> int =
  <fun>
val resolution_rule : 'a formula list list -> 'a formula list list =
  <fun>
val dp : 'a formula list list -> bool = <fun>
val dpsat : prop formula -> bool = <fun>
val dptaut : prop formula -> bool = <fun>
- : dummy_interactive = START_INTERACTIVE
// dp.p001
- : bool = true
// dp.p002
- : bool = true
- : dummy_interactive = END_INTERACTIVE
val posneg_count : 'a formula list list -> 'a formula -> int = <fun>
val dpll : 'a formula list list -> bool = <fun>
val dpllsat : prop formula -> bool = <fun>
val dplltaut : prop formula -> bool = <fun>
- : dummy_interactive = START_INTERACTIVE
// dp.p003
- : bool = true
- : dummy_interactive = END_INTERACTIVE
type trailmix = Guessed | Deduced
val unassigned :
  'a formula list list -> ('a formula * 'b) list -> 'a formula list =
  <fun>
val unit_subpropagate :
  'a formula list list * ('a formula, unit) func *
  ('a formula * trailmix) list ->
  'a formula list list * ('a formula, unit) func *
  ('a formula * trailmix) list = <fun>
val unit_propagate :
  'a formula list list * ('a formula * trailmix) list ->
  'a formula list list * ('a formula * trailmix) list = <fun>
val backtrack : ('a * trailmix) list -> ('a * trailmix) list = <fun>
val dpli : 'a formula list list -> ('a formula * trailmix) list -> bool =
  <fun>
val dplisat : prop formula -> bool = <fun>
val dplitaut : prop formula -> bool = <fun>
val backjump :
  'a formula list list ->
  'a formula ->
  ('a formula * trailmix) list -> ('a formula * trailmix) list = <fun>
```

```
val dplb : 'a formula list list -> ('a formula * trailmix) list -> bool =
  <fun>
val dplbsat : prop formula -> bool = <fun>
val dplbtaut : prop formula -> bool = <fun>
- : dummy_interactive = START_INTERACTIVE
// dp.p004
- : bool = true
// dp.p005
- : bool = true
- : dummy_interactive = END_INTERACTIVE
#
```

stal.ml

```
#use "stal.ml";;
```

```
# #use "stal.ml";;
val triplicate : prop formula -> prop formula * prop formula list =
  <fun>
val atom : 'a formula -> 'a formula = <fun>
val align : 'a formula * 'a formula -> 'a formula * 'a formula = <fun>
val equate2 :
  'a formula * 'a formula ->
  'a formula partition -> 'a formula partition = <fun>
val irredundant :
  'a formula partition ->
  ('a formula * 'a formula) list -> ('a formula * 'a formula) list =
  <fun>
val consequences :
  'a formula * 'a formula ->
  'a formula ->
  ('a formula * 'a formula) list -> ('a formula * 'a formula) list =
  <fun>
val triggers :
  'a formula ->
  (('a formula * 'a formula) * ('a formula * 'a formula) list) list =
  <fun>
- : dummy_interactive = START_INTERACTIVE
// stal.p001
- : ((prop formula * prop formula) * (prop formula * prop formula) list)
    list
=
[((<<p>>, <<true>>), [(<<q>>, <<true>>); (<<r>>, <<true>>)]);
 ((<<q>>, <<true>>), [(<<r>>, <<p>>)]);
 ((<<q>>, <<~true>>), [(<<p>>, <<~true>>)]);
 ((<<q>>, <<~p>>), [(<<p>>, <<~true>>); (<<r>>, <<p>>)]);
 ((<<r>>, <<true>>), [(<<q>>, <<p>>)]);
 ((<<r>>, <<q>>), [(<<q>>, <<p>>)]);
 ((<<r>>, <<~true>>), [(<<p>>, <<~true>>)]);
 ((<<r>>, <<~p>>), [(<<p>>, <<~true>>); (<<q>>, <<p>>)]);
 ((<<r>>, <<~q>>), [(<<p>>, <<~true>>)])]
- : dummy_interactive = END_INTERACTIVE
File "/home/student/Harrison/OCaml/stal.ml", line 66, characters 6-7:
Warning Y: unused variable p.
File "/home/student/Harrison/OCaml/stal.ml", line 66, characters 20-21:
Warning Y: unused variable q.
File "/home/student/Harrison/OCaml/stal.ml", line 66, characters 34-35:
Warning Y: unused variable r.
File "/home/student/Harrison/OCaml/stal.ml", line 63, characters 6-45:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
[]
File "/home/student/Harrison/OCaml/stal.ml", line 68, characters 14-104:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
```

```
[]
File "/home/student/Harrison/OCaml/stal.ml", line 74, characters 2-245:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
(Exists (_, _)|Forall (_, _)|Imp (_, _)|Or (_, _)|And (_, _)|Not _|Atom _|
True|False)
val trigger :
  prop formula ->
  ((prop formula * prop formula) * (prop formula * prop formula) list)
  list = <fun>
val relevance :
  (('a * 'a) * 'b) list -> ('a, (('a * 'a) * 'b) list) func = <fun>
val equatecons :
  'a formula * 'a formula ->
  'a formula partition * ('a formula, ('b * 'c list) list) func ->
  'c list *
  ('a formula partition * ('a formula, ('b * 'c list) list) func) =
  <fun>
val zero_saturate :
  'a formula partition *
  ('a formula, ('b * ('a formula * 'a formula) list) list) func ->
  ('a formula * 'a formula) list ->
  'a formula partition *
  ('a formula, ('b * ('a formula * 'a formula) list) list) func = <fun>
val zero_saturate_and_check :
  'a formula partition *
  ('a formula, ('b * ('a formula * 'a formula) list) list) func ->
  ('a formula * 'a formula) list ->
  'a formula partition *
  ('a formula, ('b * ('a formula * 'a formula) list) list) func = <fun>
val truefalse : 'a formula partition -> bool = <fun>
val equateset :
  'a formula list ->
  'a formula partition * ('a formula, ('b * 'c list) list) func ->
  'a formula partition * ('a formula, ('b * 'c list) list) func = <fun>
val inter :
  'a formula list ->
  'a formula partition * 'b ->
  'a formula partition * 'c ->
  ('a formula, 'a formula list) func ->
  ('a formula, 'a formula list) func ->
  'a formula partition * ('a formula, ('d * 'e list) list) func ->
  'a formula partition * ('a formula, ('d * 'e list) list) func = <fun>
val reverseq : 'a list -> 'a partition -> ('a, 'a list) func = <fun>
val stal_intersect :
  'a formula partition * ('a formula, ('b * 'c list) list) func ->
  'a formula partition * ('a formula, ('b * 'c list) list) func ->
  'a formula partition * ('a formula, ('b * 'c list) list) func ->
  'a formula partition * ('a formula, ('b * 'c list) list) func = <fun>
val saturate :
  int ->
  'a formula partition *
  ('a formula, ('b * ('a formula * 'a formula) list) list) func ->
  ('a formula * 'a formula) list ->
  'a formula list ->
  'a formula partition *
  ('a formula, ('b * ('a formula * 'a formula) list) list) func = <fun>
val splits :
  int ->
  'a formula partition *
  ('a formula, ('b * ('a formula * 'a formula) list) list) func ->
  'a formula list ->
  'a formula list ->
  'a formula partition *
  ('a formula, ('b * ('a formula * 'a formula) list) list) func = <fun>
val saturate_upto :
  'a formula list ->
  int ->
  int ->
  (('a formula * 'a formula) * ('a formula * 'a formula) list) list ->
  ('a formula * 'a formula) list -> bool = <fun>
val stalmarck : prop formula -> bool = <fun>
- : dummy_interactive = START_INTERACTIVE
// stal.p002
*** Starting 0-saturation
*** Starting 1-saturation
*** Starting 2-saturation
CPU time (user): 9.92862
- : bool = true
- : dummy_interactive = END_INTERACTIVE
#
```

bdd.ml

```
#use "bdd.ml";;
```

```
# #use "bdd.ml";;
type bddnode = prop * int * int
type bdd =
    Bdd of ((bddnode, int) func * (int, bddnode) func * int) *
      (prop -> prop -> bool)
val print_bdd : bdd -> unit = <fun>
val expand_node : bdd -> int -> bddnode = <fun>
val lookup_unique : bdd -> bddnode -> bdd * int = <fun>
val mk_node : bdd -> prop * int * int -> bdd * int = <fun>
val mk_bdd : (prop -> prop -> bool) -> bdd = <fun>
val order : bdd -> prop -> prop -> bool = <fun>
val thread :
  'a ->
  ('b -> 'c * 'd -> 'e) ->
  ('a -> 'f -> 'g * 'c) * 'f -> ('g -> 'h -> 'b * 'd) * 'h -> 'e =
  <fun>
val bdd_and :
  bdd * (int * int, int) func ->
  int * int -> (bdd * (int * int, int) func) * int = <fun>
val bdd_or :
  bdd * (int * int, int) func ->
  int * int -> (bdd * (int * int, int) func) * int = <fun>
val bdd_imp :
  bdd * (int * int, int) func ->
  int * int -> (bdd * (int * int, int) func) * int = <fun>
val bdd_iff :
  bdd * (int * int, int) func ->
  int * int -> (bdd * (int * int, int) func) * int = <fun>
File "/home/student/Harrison/OCaml/bdd.ml", line 104, characters 2-423:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
(Exists (_, _)|Forall (_, _))
val mkbdd :
  bdd * (int * int, int) func ->
  prop formula -> (bdd * (int * int, int) func) * int = <fun>
val bddtaut : prop formula -> bool = <fun>
- : dummy_interactive = START_INTERACTIVE
// bdd.p001
- : bool = true
- : dummy_interactive = END_INTERACTIVE
val dest_nimp : 'a formula -> 'a formula * 'a formula = <fun>
val dest_iffdef : 'a formula -> 'a * 'a formula = <fun>
val restore_iffdef : 'a * 'a formula -> 'a formula -> 'a formula =
  <fun>
val suitable_iffdef : ('a * 'b) list -> 'c * 'a formula -> bool = <fun>
val sort_defs :
  ('a * 'a formula) list ->
  ('a * 'a formula) list ->
  'a formula -> ('a * 'a formula) list * 'a formula = <fun>
```

```
File "/home/student/Harrison/OCaml/bdd.ml", line 156, characters 2-528:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
(Exists (_, _)|Forall (_, _))
val mkbdde :
  (prop, int) func ->
  bdd * (int * int, int) func ->
  prop formula -> (bdd * (int * int, int) func) * int = <fun>
val mkbdds :
  (prop, int) func ->
  bdd * (int * int, int) func ->
  (prop * prop formula) list ->
  prop formula -> (bdd * (int * int, int) func) * int = <fun>
val ebddtaut : prop formula -> bool = <fun>
- : dummy_interactive = START_INTERACTIVE
// bdd.p002
- : bool = true
// bdd.p003
- : bool = true
- : dummy_interactive = END_INTERACTIVE
#
```

fol.ml

```
#use "fol.ml";;
```

```
# #use "fol.ml";;
type term = Var of string | Fn of string * term list
- : dummy_interactive = START_INTERACTIVE
// fol.p001
- : term =
Fn ("sqrt",
 [Fn ("-",
   [Fn ("1", []);
    Fn ("cos",
     [Fn ("power", [Fn ("+", [Var "x"; Var "y"]); Fn ("2", [])])])])])
- : dummy_interactive = END_INTERACTIVE
type fol = R of string * term list
val onformula : (term -> term) -> fol formula -> fol formula = <fun>
- : dummy_interactive = START_INTERACTIVE
// fol.p002
- : fol formula =
Atom (R ("<", [Fn ("+", [Var "x"; Var "y"]); Var "z"]))
- : dummy_interactive = END_INTERACTIVE
val is_const_name : string -> bool = <fun>
val parse_atomic_term :
  string list -> string list -> term * string list = <fun>
val parse_term : string list -> string list -> term * string list =
  <fun>
val parset : string -> term = <fun>
val parse_infix_atom :
  string list -> string list -> fol formula * string list = <fun>
val parse_atom :
  string list -> string list -> fol formula * string list = <fun>
val parse : string -> fol formula = <fun>
val default_parser : string -> fol formula = <fun>
val secondary_parser : string -> term = <fun>
- : dummy_interactive = START_INTERACTIVE
// fol.p003
- : fol formula =
Or
 (Forall ("x",
   Imp (Atom (R ("<", [Var "x"; Fn ("2", [])])),
    Atom (R ("<=", [Fn ("*", [Fn ("2", []); Var "x"]); Fn ("3", [])])))),
 False)
// fol.p004
- : term = Fn ("*", [Fn ("2", []); Var "x"])
- : dummy_interactive = END_INTERACTIVE
val print_term : int -> term -> unit = <fun>
val print_fargs : string -> term list -> unit = <fun>
val print_infix_term :
  bool -> int -> int -> string -> term -> term -> unit = <fun>
val printert : term -> unit = <fun>
val print_atom : 'a -> fol -> unit = <fun>
val print_fol_formula : fol formula -> unit = <fun>
```

```
- : dummy_interactive = START_INTERACTIVE
// fol.p005
- : fol formula = <<forall x y. exists z. x < z /\ y < z>>
// fol.p006
- : fol formula = <<~(forall x. P(x)) <=> (exists y. ~P(y))>>
- : dummy_interactive = END_INTERACTIVE
val termval :
  'a * (string -> 'b list -> 'b) * 'c ->
  (string, 'b) func -> term -> 'b = <fun>
val holds :
  'a list * (string -> 'a list -> 'a) * (string -> 'a list -> bool) ->
  (string, 'a) func -> fol formula -> bool = <fun>
val bool_interp :
  bool list * (string -> bool list -> bool) *
  (string -> 'a list -> bool) = ([false; true], <fun>, <fun>)
val mod_interp :
  int ->
  int list * (string -> int list -> int) * (string -> 'a list -> bool) =
  <fun>
- : dummy_interactive = START_INTERACTIVE
// fol.p007
- : bool = true
// fol.p008
- : bool = true
// fol.p009
- : bool = false
val fm : fol formula = <<forall x. ~x = 0 ==> (exists y. x * y = 1)>>
// fol.p010
- : int list = [1; 2; 3; 5; 7; 11; 13; 17; 19; 23; 29; 31; 37; 41; 43]
// fol.p011
- : bool = true
// fol.p012
- : bool = false
- : dummy_interactive = END_INTERACTIVE
val fvt : term -> string list = <fun>
val var : fol formula -> string list = <fun>
val fv : fol formula -> string list = <fun>
val generalize : fol formula -> fol formula = <fun>
val tsubst : (string, term) func -> term -> term = <fun>
val variant : string -> string list -> string = <fun>
- : dummy_interactive = START_INTERACTIVE
// fol.p013
- : string = "x"
// fol.p014
- : string = "x'"
// fol.p015
- : string = "x''"
- : dummy_interactive = END_INTERACTIVE
val subst : (string, term) func -> fol formula -> fol formula = <fun>
val substq :
  (string, term) func ->
  (string -> fol formula -> fol formula) ->
  string -> fol formula -> fol formula = <fun>
- : dummy_interactive = START_INTERACTIVE
// fol.p016
```

```
- : fol formula = <<forall x'. x' = x>>
// fol.p017
- : fol formula = <<forall x' x''. x' = x ==> x' = x''>>
- : dummy_interactive = END_INTERACTIVE
#
```

skolem.ml

```
#use "skolem.ml";;
```

```
# #use "skolem.ml";;
val simplify1 : fol formula -> fol formula = <fun>
val simplify : fol formula -> fol formula = <fun>
- : dummy_interactive = START_INTERACTIVE
// skolem.p001
- : fol formula = <<(forall x. P(x)) ==> Q>>
- : dummy_interactive = END_INTERACTIVE
val nnf : 'a formula -> 'a formula = <fun>
- : dummy_interactive = START_INTERACTIVE
// skolem.p002
- : fol formula =
<<(exists x. ~P(x)) \/
  (exists y. Q(y)) /\ (exists z. P(z) /\ Q(z)) \/
  (forall y. ~Q(y)) /\ (forall z. ~P(z) \/ ~Q(z))>>
- : dummy_interactive = END_INTERACTIVE
val pullquants : fol formula -> fol formula = <fun>
val pullq :
  bool * bool ->
  fol formula ->
  (string -> fol formula -> fol formula) ->
  (fol formula -> fol formula -> fol formula) ->
  string -> string -> fol formula -> fol formula -> fol formula = <fun>
val prenex : fol formula -> fol formula = <fun>
val pnf : fol formula -> fol formula = <fun>
- : dummy_interactive = START_INTERACTIVE
// skolem.p003
- : fol formula =
<<exists x. forall z. ~P(x) /\ ~R(y) \/ Q(x) \/ ~P(z) \/ ~Q(z)>>
- : dummy_interactive = END_INTERACTIVE
val funcs : term -> (string * int) list = <fun>
val functions : fol formula -> (string * int) list = <fun>
val skolem : fol formula -> string list -> fol formula * string list =
  <fun>
val skolem2 :
  (fol formula * fol formula -> fol formula) ->
  fol formula * fol formula -> string list -> fol formula * string list =
  <fun>
val askolemize : fol formula -> fol formula = <fun>
val specialize : 'a formula -> 'a formula = <fun>
val skolemize : fol formula -> fol formula = <fun>
- : dummy_interactive = START_INTERACTIVE
// skolem.p004
- : fol formula = <<~x < f_y(x) \/ x * u < f_y(x) * f_v(u,x)>>
// skolem.p005
- : fol formula = <<~P(x) \/ Q(c_y) \/ ~P(z) \/ ~Q(z)>>
- : dummy_interactive = END_INTERACTIVE
#
```

herbrand.ml

```
#use "herbrand.ml";;
```

```
# #use "herbrand.ml";;
val pholds : ('a formula -> bool) -> 'a formula -> bool = <fun>
val herbfuns : fol formula -> (string * int) list * (string * int) list =
  <fun>
val groundterms : term list -> (string * int) list -> int -> term list =
  <fun>
val groundtuples :
  term list -> (string * int) list -> int -> int -> term list list =
  <fun>
val herbloop :
  ('a -> (fol formula -> fol formula) -> 'b list -> 'b list) ->
  ('b list -> bool) ->
  'a ->
  term list ->
  (string * int) list ->
  string list ->
  int -> 'b list -> term list list -> term list list -> term list list =
  <fun>
val gilmore_loop :
  fol formula list list ->
  term list ->
  (string * int) list ->
  string list ->
  int ->
  fol formula list list ->
  term list list -> term list list -> term list list = <fun>
val gilmore : fol formula -> int = <fun>
- : dummy_interactive = START_INTERACTIVE
// herbrand.p001
0 ground instances tried; 1 items in list
0 ground instances tried; 1 items in list
1 ground instances tried; 1 items in list
1 ground instances tried; 1 items in list
- : int = 2
// herbrand.p002
val sfm : fol formula = <<P(x) /\ ~P(f_y(x))>>
// herbrand.p003
0 ground instances tried; 1 items in list
0 ground instances tried; 1 items in list
val p24 : int = 1
// herbrand.p004
0 ground instances tried; 1 items in list
0 ground instances tried; 1 items in list
1 ground instances tried; 13 items in list
1 ground instances tried; 13 items in list
2 ground instances tried; 57 items in list
3 ground instances tried; 84 items in list
4 ground instances tried; 405 items in list
val p45 : int = 5
```

```
- : dummy_interactive = END_INTERACTIVE
val dp_mfn : 'a list list -> ('a -> 'b) -> 'b list list -> 'b list list =
  <fun>
val dp_loop :
  fol formula list list ->
  term list ->
  (string * int) list ->
  string list ->
  int ->
  fol formula list list ->
  term list list -> term list list -> term list list = <fun>
val davisputnam : fol formula -> int = <fun>
- : dummy_interactive = START_INTERACTIVE
// herbrand.p006
0 ground instances tried; 0 items in list
0 ground instances tried; 0 items in list
1 ground instances tried; 5 items in list
2 ground instances tried; 7 items in list
3 ground instances tried; 10 items in list
4 ground instances tried; 12 items in list
5 ground instances tried; 13 items in list
6 ground instances tried; 14 items in list
7 ground instances tried; 15 items in list
8 ground instances tried; 16 items in list
8 ground instances tried; 16 items in list
9 ground instances tried; 18 items in list
10 ground instances tried; 20 items in list
11 ground instances tried; 22 items in list
12 ground instances tried; 24 items in list
13 ground instances tried; 26 items in list
14 ground instances tried; 28 items in list
15 ground instances tried; 30 items in list
16 ground instances tried; 32 items in list
17 ground instances tried; 35 items in list
18 ground instances tried; 37 items in list
val p20 : int = 19
- : dummy_interactive = END_INTERACTIVE
val dp_refine :
  fol formula list list ->
  string list -> term list list -> term list list -> term list list =
  <fun>
val dp_refine_loop :
  fol formula list list ->
  term list ->
  (string * int) list ->
  string list ->
  int ->
  fol formula list list ->
  term list list -> term list list -> term list list = <fun>
val davisputnam' : fol formula -> int = <fun>
- : dummy_interactive = START_INTERACTIVE
// herbrand.p007
0 ground instances tried; 0 items in list
0 ground instances tried; 0 items in list
1 ground instances tried; 6 items in list
```

```
1 ground instances tried; 6 items in list
2 ground instances tried; 10 items in list
3 ground instances tried; 14 items in list
4 ground instances tried; 18 items in list
5 ground instances tried; 22 items in list
6 ground instances tried; 29 items in list
7 ground instances tried; 36 items in list
7 ground instances tried; 36 items in list
8 ground instances tried; 40 items in list
9 ground instances tried; 44 items in list
10 ground instances tried; 48 items in list
11 ground instances tried; 52 items in list
12 ground instances tried; 56 items in list
13 ground instances tried; 60 items in list
14 ground instances tried; 64 items in list
15 ground instances tried; 68 items in list
16 ground instances tried; 72 items in list
17 ground instances tried; 76 items in list
18 ground instances tried; 80 items in list
19 ground instances tried; 84 items in list
20 ground instances tried; 88 items in list
21 ground instances tried; 92 items in list
22 ground instances tried; 96 items in list
23 ground instances tried; 100 items in list
24 ground instances tried; 104 items in list
25 ground instances tried; 108 items in list
26 ground instances tried; 112 items in list
27 ground instances tried; 116 items in list
28 ground instances tried; 123 items in list
29 ground instances tried; 130 items in list
30 ground instances tried; 137 items in list
31 ground instances tried; 144 items in list
31 ground instances tried; 144 items in list
32 ground instances tried; 148 items in list
33 ground instances tried; 152 items in list
34 ground instances tried; 156 items in list
35 ground instances tried; 160 items in list
36 ground instances tried; 164 items in list
37 ground instances tried; 168 items in list
38 ground instances tried; 172 items in list
39 ground instances tried; 176 items in list
val p36 : int = 3
// herbrand.p008
0 ground instances tried; 0 items in list
0 ground instances tried; 0 items in list
1 ground instances tried; 31 items in list
2 ground instances tried; 41 items in list
3 ground instances tried; 51 items in list
4 ground instances tried; 61 items in list
5 ground instances tried; 71 items in list
6 ground instances tried; 81 items in list
7 ground instances tried; 91 items in list
8 ground instances tried; 101 items in list
9 ground instances tried; 111 items in list
10 ground instances tried; 121 items in list
```

```
11 ground instances tried; 131 items in list
12 ground instances tried; 141 items in list
13 ground instances tried; 151 items in list
14 ground instances tried; 161 items in list
15 ground instances tried; 171 items in list
16 ground instances tried; 181 items in list
17 ground instances tried; 191 items in list
18 ground instances tried; 201 items in list
19 ground instances tried; 211 items in list
20 ground instances tried; 221 items in list
21 ground instances tried; 231 items in list
22 ground instances tried; 241 items in list
23 ground instances tried; 251 items in list
24 ground instances tried; 261 items in list
25 ground instances tried; 271 items in list
26 ground instances tried; 281 items in list
27 ground instances tried; 291 items in list
28 ground instances tried; 301 items in list
29 ground instances tried; 311 items in list
30 ground instances tried; 321 items in list
31 ground instances tried; 331 items in list
32 ground instances tried; 341 items in list
33 ground instances tried; 351 items in list
34 ground instances tried; 361 items in list
35 ground instances tried; 371 items in list
36 ground instances tried; 381 items in list
37 ground instances tried; 391 items in list
38 ground instances tried; 393 items in list
39 ground instances tried; 396 items in list
40 ground instances tried; 399 items in list
41 ground instances tried; 402 items in list
42 ground instances tried; 405 items in list
43 ground instances tried; 407 items in list
44 ground instances tried; 410 items in list
45 ground instances tried; 414 items in list
46 ground instances tried; 418 items in list
47 ground instances tried; 422 items in list
48 ground instances tried; 426 items in list
49 ground instances tried; 429 items in list
50 ground instances tried; 433 items in list
51 ground instances tried; 437 items in list
52 ground instances tried; 441 items in list
53 ground instances tried; 445 items in list
54 ground instances tried; 449 items in list
55 ground instances tried; 452 items in list
56 ground instances tried; 456 items in list
57 ground instances tried; 460 items in list
58 ground instances tried; 464 items in list
59 ground instances tried; 468 items in list
60 ground instances tried; 472 items in list
61 ground instances tried; 475 items in list
62 ground instances tried; 479 items in list
63 ground instances tried; 483 items in list
64 ground instances tried; 487 items in list
65 ground instances tried; 491 items in list
```

```
66 ground instances tried; 495 items in list
67 ground instances tried; 498 items in list
68 ground instances tried; 502 items in list
69 ground instances tried; 506 items in list
70 ground instances tried; 510 items in list
71 ground instances tried; 514 items in list
72 ground instances tried; 518 items in list
73 ground instances tried; 528 items in list
74 ground instances tried; 530 items in list
75 ground instances tried; 532 items in list
76 ground instances tried; 535 items in list
77 ground instances tried; 538 items in list
78 ground instances tried; 541 items in list
79 ground instances tried; 543 items in list
80 ground instances tried; 545 items in list
81 ground instances tried; 547 items in list
82 ground instances tried; 550 items in list
83 ground instances tried; 553 items in list
84 ground instances tried; 556 items in list
85 ground instances tried; 558 items in list
86 ground instances tried; 560 items in list
87 ground instances tried; 563 items in list
88 ground instances tried; 567 items in list
89 ground instances tried; 571 items in list
90 ground instances tried; 575 items in list
91 ground instances tried; 578 items in list
92 ground instances tried; 581 items in list
93 ground instances tried; 585 items in list
94 ground instances tried; 589 items in list
95 ground instances tried; 593 items in list
96 ground instances tried; 597 items in list
97 ground instances tried; 600 items in list
98 ground instances tried; 603 items in list
99 ground instances tried; 607 items in list
100 ground instances tried; 611 items in list
101 ground instances tried; 615 items in list
102 ground instances tried; 619 items in list
103 ground instances tried; 622 items in list
104 ground instances tried; 625 items in list
105 ground instances tried; 629 items in list
106 ground instances tried; 633 items in list
107 ground instances tried; 637 items in list
108 ground instances tried; 641 items in list
109 ground instances tried; 651 items in list
110 ground instances tried; 653 items in list
111 ground instances tried; 655 items in list
112 ground instances tried; 657 items in list
113 ground instances tried; 660 items in list
114 ground instances tried; 663 items in list
115 ground instances tried; 665 items in list
116 ground instances tried; 667 items in list
117 ground instances tried; 669 items in list
118 ground instances tried; 671 items in list
119 ground instances tried; 674 items in list
120 ground instances tried; 677 items in list
```

```
121 ground instances tried; 679 items in list
122 ground instances tried; 681 items in list
123 ground instances tried; 683 items in list
124 ground instances tried; 685 items in list
125 ground instances tried; 688 items in list
126 ground instances tried; 691 items in list
127 ground instances tried; 693 items in list
128 ground instances tried; 695 items in list
129 ground instances tried; 697 items in list
130 ground instances tried; 700 items in list
131 ground instances tried; 704 items in list
132 ground instances tried; 708 items in list
133 ground instances tried; 711 items in list
134 ground instances tried; 714 items in list
135 ground instances tried; 717 items in list
136 ground instances tried; 721 items in list
137 ground instances tried; 725 items in list
138 ground instances tried; 729 items in list
139 ground instances tried; 732 items in list
140 ground instances tried; 735 items in list
141 ground instances tried; 738 items in list
142 ground instances tried; 742 items in list
143 ground instances tried; 746 items in list
144 ground instances tried; 750 items in list
145 ground instances tried; 760 items in list
146 ground instances tried; 762 items in list
147 ground instances tried; 764 items in list
148 ground instances tried; 766 items in list
149 ground instances tried; 768 items in list
150 ground instances tried; 771 items in list
151 ground instances tried; 773 items in list
152 ground instances tried; 775 items in list
153 ground instances tried; 777 items in list
154 ground instances tried; 779 items in list
155 ground instances tried; 781 items in list
156 ground instances tried; 784 items in list
157 ground instances tried; 786 items in list
158 ground instances tried; 788 items in list
159 ground instances tried; 790 items in list
160 ground instances tried; 792 items in list
161 ground instances tried; 794 items in list
162 ground instances tried; 797 items in list
163 ground instances tried; 799 items in list
164 ground instances tried; 801 items in list
165 ground instances tried; 803 items in list
166 ground instances tried; 805 items in list
167 ground instances tried; 807 items in list
168 ground instances tried; 810 items in list
169 ground instances tried; 812 items in list
170 ground instances tried; 814 items in list
171 ground instances tried; 816 items in list
172 ground instances tried; 818 items in list
173 ground instances tried; 821 items in list
174 ground instances tried; 825 items in list
175 ground instances tried; 828 items in list
```

```
176 ground instances tried; 831 items in list
177 ground instances tried; 834 items in list
178 ground instances tried; 837 items in list
179 ground instances tried; 841 items in list
180 ground instances tried; 845 items in list
val p29 : int = 5
- : dummy_interactive = END_INTERACTIVE
#
```

unif.ml

```
#use "unif.ml";;
```

```
# #use "unif.ml";;
val istriv : (string, term) func -> string -> term -> bool = <fun>
val unify :
  (string, term) func -> (term * term) list -> (string, term) func =
  <fun>
val solve : (string, term) func -> (string, term) func = <fun>
val fullunify : (term * term) list -> (string, term) func = <fun>
val unify_and_apply : (term * term) list -> (term * term) list = <fun>
- : dummy_interactive = START_INTERACTIVE
// unify.p001
- : (term * term) list = [(<<|f(f(z),g(y))|>>, <<|f(f(z),g(y))|>>)]
// unify.p002
- : (term * term) list = [(<<|f(y,y)|>>, <<|f(y,y)|>>)]
// unify.p004
- : (term * term) list =
[(<<|f(f(f(x_3,x_3),f(x_3,x_3)),f(f(x_3,x_3),f(x_3,x_3)))|>>,
  <<|f(f(f(x_3,x_3),f(x_3,x_3)),f(f(x_3,x_3),f(x_3,x_3)))|>>);
 (<<|f(f(x_3,x_3),f(x_3,x_3))|>>, <<|f(f(x_3,x_3),f(x_3,x_3))|>>);
 (<<|f(x_3,x_3)|>>, <<|f(x_3,x_3)|>>)]
- : dummy_interactive = END_INTERACTIVE
#
```

tableaux.ml

```
#use "tableaux.ml";;
```

```
# #use "tableaux.ml";;
val unify_literals :
  (string, term) func ->
  fol formula * fol formula -> (string, term) func = <fun>
val unify_complements :
  (string, term) func ->
  fol formula * fol formula -> (string, term) func = <fun>
val unify_refute :
  fol formula list list -> (string, term) func -> (string, term) func =
  <fun>
val prawitz_loop :
  fol formula list list ->
  string list ->
  fol formula list list -> int -> (string, term) func * int = <fun>
val prawitz : fol formula -> int = <fun>
- : dummy_interactive = START_INTERACTIVE
// tableaux.p001
val p20 : int = 2
- : dummy_interactive = END_INTERACTIVE
val compare : fol formula -> int * int = <fun>
- : dummy_interactive = START_INTERACTIVE
// tableaux.p002
0 ground instances tried; 0 items in list
0 ground instances tried; 0 items in list
1 ground instances tried; 3 items in list
1 ground instances tried; 3 items in list
2 ground instances tried; 6 items in list
val p19 : int * int = (3, 3)
// tableaux.p003
0 ground instances tried; 0 items in list
0 ground instances tried; 0 items in list
1 ground instances tried; 5 items in list
2 ground instances tried; 7 items in list
3 ground instances tried; 10 items in list
4 ground instances tried; 12 items in list
5 ground instances tried; 13 items in list
6 ground instances tried; 14 items in list
7 ground instances tried; 15 items in list
8 ground instances tried; 16 items in list
8 ground instances tried; 16 items in list
9 ground instances tried; 18 items in list
10 ground instances tried; 20 items in list
11 ground instances tried; 22 items in list
12 ground instances tried; 24 items in list
13 ground instances tried; 26 items in list
14 ground instances tried; 28 items in list
15 ground instances tried; 30 items in list
16 ground instances tried; 32 items in list
17 ground instances tried; 35 items in list
```

```
18 ground instances tried; 37 items in list
val p20 : int * int = (2, 19)
// tableaux.p004
0 ground instances tried; 0 items in list
0 ground instances tried; 0 items in list
val p24 : int * int = (1, 1)
// tableaux.p005
0 ground instances tried; 0 items in list
0 ground instances tried; 0 items in list
val p39 : int * int = (1, 1)
// tableaux.p006
0 ground instances tried; 0 items in list
0 ground instances tried; 0 items in list
1 ground instances tried; 5 items in list
1 ground instances tried; 5 items in list
2 ground instances tried; 8 items in list
val p42 : int * int = (2, 3)
// tableaux.p008
0 ground instances tried; 0 items in list
0 ground instances tried; 0 items in list
1 ground instances tried; 7 items in list
1 ground instances tried; 7 items in list
2 ground instances tried; 13 items in list
val p44 : int * int = (2, 3)
// tableaux.p009
0 ground instances tried; 0 items in list
0 ground instances tried; 0 items in list
1 ground instances tried; 3 items in list
1 ground instances tried; 3 items in list
val p59 : int * int = (2, 2)
// tableaux.p010
0 ground instances tried; 0 items in list
0 ground instances tried; 0 items in list
1 ground instances tried; 8 items in list
2 ground instances tried; 11 items in list
3 ground instances tried; 17 items in list
4 ground instances tried; 19 items in list
4 ground instances tried; 19 items in list
5 ground instances tried; 22 items in list
6 ground instances tried; 25 items in list
7 ground instances tried; 28 items in list
8 ground instances tried; 31 items in list
9 ground instances tried; 33 items in list
10 ground instances tried; 35 items in list
11 ground instances tried; 37 items in list
12 ground instances tried; 39 items in list
val p60 : int * int = (1, 13)
- : dummy_interactive = END_INTERACTIVE
val tableau :
  fol formula list * fol formula list * int ->
  ((string, term) func * int -> 'a) -> (string, term) func * int -> 'a =
  <fun>
val deepen : (int -> 'a) -> int -> 'a = <fun>
File "/home/student/Harrison/OCaml/tableaux.ml", line 138, characters 19-64:
Warning S: this expression should have type unit.
```

```
val tabrefute : fol formula list -> int = <fun>
val tab : fol formula -> int = <fun>
- : dummy_interactive = START_INTERACTIVE
// tableaux.p011
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
val p38 : int = 4
- : dummy_interactive = END_INTERACTIVE
val splittab : fol formula -> int list = <fun>
- : dummy_interactive = START_INTERACTIVE
// tableaux.p012
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
val p34 : int list =
  [5; 4; 5; 3; 3; 3; 2; 4; 6; 2; 3; 3; 4; 3; 3; 3; 3; 2; 2; 3; 6; 3; 2;
   4; 3; 3; 3; 3; 3; 4; 4; 4]
// tableaux.p013
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
Searching with depth limit 8
Searching with depth limit 9
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
Searching with depth limit 8
Searching with depth limit 9
val ewd1062 : int list = [9; 9]
- : dummy_interactive = END_INTERACTIVE
#
```

resolution.ml

```
#use "resolution.ml";;
```

```
# #use "resolution.ml";;
val barb : fol formula =
  <<~(exists b. forall x. shaves(b,x) <=> ~shaves(x,x))>>
- : dummy_interactive = START_INTERACTIVE
// resolution.p001
- : fol formula list list =
[[<<shaves(x,x)>>; <<shaves(c_b,x)>>];
 [<<~shaves(x,x)>>; <<~shaves(c_b,x)>>]]
- : dummy_interactive = END_INTERACTIVE
val mgu :
  fol formula list -> (string, term) func -> (string, term) func =
  <fun>
val unifiable : fol formula -> fol formula -> bool = <fun>
val rename : string -> fol formula list -> fol formula list = <fun>
val resolvents :
  fol formula list ->
  fol formula list ->
  fol formula -> fol formula list list -> fol formula list list = <fun>
val resolve_clauses :
  fol formula list -> fol formula list -> fol formula list list = <fun>
val resloop : fol formula list list * fol formula list list -> bool =
  <fun>
val pure_resolution : fol formula -> bool = <fun>
val resolution : fol formula -> bool list = <fun>
- : dummy_interactive = START_INTERACTIVE
// resolution.p002
0 used; 3 unused.
1 used; 2 unused.
2 used; 3 unused.
3 used; 6 unused.
4 used; 8 unused.
5 used; 10 unused.
6 used; 16 unused.
7 used; 22 unused.
8 used; 28 unused.
9 used; 34 unused.
10 used; 41 unused.
11 used; 47 unused.
12 used; 53 unused.
13 used; 62 unused.
14 used; 69 unused.
15 used; 76 unused.
16 used; 85 unused.
17 used; 92 unused.
18 used; 99 unused.
19 used; 105 unused.
20 used; 111 unused.
21 used; 117 unused.
22 used; 123 unused.
```

```
23 used; 132 unused.
24 used; 139 unused.
25 used; 146 unused.
26 used; 152 unused.
27 used; 158 unused.
28 used; 164 unused.
29 used; 170 unused.
30 used; 177 unused.
31 used; 184 unused.
32 used; 191 unused.
33 used; 198 unused.
34 used; 203 unused.
35 used; 207 unused.
36 used; 211 unused.
37 used; 218 unused.
37 used; 225 unused.
38 used; 232 unused.
39 used; 239 unused.
40 used; 244 unused.
41 used; 248 unused.
42 used; 252 unused.
43 used; 255 unused.
44 used; 260 unused.
45 used; 265 unused.
46 used; 268 unused.
47 used; 274 unused.
48 used; 280 unused.
49 used; 285 unused.
50 used; 290 unused.
51 used; 296 unused.
52 used; 302 unused.
53 used; 308 unused.
54 used; 312 unused.
55 used; 315 unused.
56 used; 318 unused.
57 used; 320 unused.
58 used; 326 unused.
59 used; 332 unused.
60 used; 338 unused.
61 used; 342 unused.
62 used; 345 unused.
63 used; 348 unused.
64 used; 350 unused.
64 used; 353 unused.
64 used; 358 unused.
64 used; 363 unused.
64 used; 366 unused.
64 used; 372 unused.
64 used; 378 unused.
65 used; 380 unused.
66 used; 382 unused.
66 used; 387 unused.
66 used; 392 unused.
67 used; 398 unused.
68 used; 404 unused.
```

```
68 used; 410 unused.
69 used; 414 unused.
70 used; 417 unused.
71 used; 420 unused.
72 used; 422 unused.
72 used; 424 unused.
73 used; 430 unused.
74 used; 436 unused.
74 used; 442 unused.
75 used; 446 unused.
76 used; 449 unused.
77 used; 452 unused.
78 used; 454 unused.
78 used; 456 unused.
79 used; 462 unused.
80 used; 468 unused.
81 used; 473 unused.
82 used; 478 unused.
83 used; 483 unused.
84 used; 488 unused.
val davis_putnam_example : bool list = [true]
- : dummy_interactive = END_INTERACTIVE
val term_match :
  (string, term) func -> (term * term) list -> (string, term) func =
  <fun>
val match_literals :
  (string, term) func ->
  fol formula * fol formula -> (string, term) func = <fun>
val subsumes_clause : fol formula list -> fol formula list -> bool =
  <fun>
val replace :
  fol formula list -> fol formula list list -> fol formula list list =
  <fun>
val incorporate :
  fol formula list ->
  fol formula list -> fol formula list list -> fol formula list list =
  <fun>
val resloop : fol formula list list * fol formula list list -> bool =
  <fun>
val pure_resolution : fol formula -> bool = <fun>
val resolution : fol formula -> bool list = <fun>
- : dummy_interactive = START_INTERACTIVE
// resolution.p003
0 used; 3 unused.
1 used; 2 unused.
2 used; 3 unused.
3 used; 6 unused.
4 used; 5 unused.
5 used; 4 unused.
6 used; 3 unused.
7 used; 2 unused.
val davis_putnam_example : bool list = [true]
- : dummy_interactive = END_INTERACTIVE
val presolve_clauses :
  fol formula list -> fol formula list -> fol formula list list = <fun>
```

```
val presloop : fol formula list list * fol formula list list -> bool =
  <fun>
val pure_presolution : fol formula -> bool = <fun>
val presolution : fol formula -> bool list = <fun>
- : dummy_interactive = START_INTERACTIVE
// resolution.p004
0 used; 6 unused.
1 used; 5 unused.
2 used; 7 unused.
3 used; 8 unused.
4 used; 7 unused.
5 used; 7 unused.
6 used; 7 unused.
7 used; 13 unused.
8 used; 14 unused.
9 used; 14 unused.
10 used; 15 unused.
11 used; 15 unused.
12 used; 18 unused.
13 used; 21 unused.
14 used; 24 unused.
15 used; 28 unused.
16 used; 33 unused.
17 used; 35 unused.
18 used; 36 unused.
19 used; 38 unused.
20 used; 39 unused.
21 used; 48 unused.
22 used; 47 unused.
23 used; 48 unused.
24 used; 47 unused.
25 used; 46 unused.
26 used; 46 unused.
27 used; 46 unused.
28 used; 46 unused.
29 used; 45 unused.
30 used; 44 unused.
31 used; 44 unused.
32 used; 44 unused.
33 used; 43 unused.
34 used; 42 unused.
35 used; 41 unused.
CPU time (user): 0.060003
val los : bool list = [true]
val pure_resolution : fol formula -> bool = <fun>
val resolution : fol formula -> bool list = <fun>
- : dummy_interactive = END_INTERACTIVE
- : dummy_interactive = START_INTERACTIVE
// resolution.p066
10 used; 1 unused.
11 used; 6 unused.
12 used; 9 unused.
13 used; 12 unused.
14 used; 13 unused.
15 used; 14 unused.
```

```
16 used; 15 unused.
17 used; 16 unused.
18 used; 22 unused.
19 used; 31 unused.
20 used; 33 unused.
21 used; 36 unused.
22 used; 36 unused.
23 used; 43 unused.
24 used; 46 unused.
25 used; 49 unused.
26 used; 49 unused.
27 used; 52 unused.
28 used; 52 unused.
29 used; 53 unused.
30 used; 57 unused.
31 used; 60 unused.
32 used; 64 unused.
33 used; 67 unused.
34 used; 68 unused.
35 used; 72 unused.
36 used; 78 unused.
37 used; 87 unused.
38 used; 91 unused.
39 used; 92 unused.
40 used; 92 unused.
41 used; 92 unused.
42 used; 99 unused.
43 used; 99 unused.
44 used; 102 unused.
45 used; 102 unused.
46 used; 107 unused.
47 used; 108 unused.
48 used; 110 unused.
49 used; 109 unused.
50 used; 110 unused.
51 used; 109 unused.
52 used; 109 unused.
53 used; 109 unused.
54 used; 108 unused.
55 used; 107 unused.
56 used; 106 unused.
57 used; 106 unused.
58 used; 105 unused.
59 used; 106 unused.
60 used; 106 unused.
60 used; 105 unused.
61 used; 105 unused.
62 used; 104 unused.
63 used; 103 unused.
64 used; 102 unused.
65 used; 101 unused.
66 used; 100 unused.
67 used; 99 unused.
68 used; 98 unused.
69 used; 97 unused.
```

```
70 used; 96 unused.
71 used; 96 unused.
72 used; 95 unused.
73 used; 94 unused.
74 used; 93 unused.
74 used; 92 unused.
74 used; 91 unused.
74 used; 90 unused.
74 used; 89 unused.
75 used; 89 unused.
76 used; 89 unused.
77 used; 88 unused.
78 used; 91 unused.
79 used; 90 unused.
80 used; 91 unused.
81 used; 94 unused.
82 used; 93 unused.
82 used; 92 unused.
83 used; 93 unused.
84 used; 95 unused.
85 used; 96 unused.
86 used; 97 unused.
87 used; 96 unused.
88 used; 96 unused.
89 used; 97 unused.
90 used; 100 unused.
91 used; 100 unused.
92 used; 100 unused.
93 used; 101 unused.
94 used; 100 unused.
95 used; 102 unused.
96 used; 101 unused.
96 used; 100 unused.
97 used; 100 unused.
98 used; 103 unused.
99 used; 102 unused.
100 used; 105 unused.
101 used; 105 unused.
102 used; 107 unused.
103 used; 109 unused.
104 used; 114 unused.
105 used; 117 unused.
106 used; 118 unused.
107 used; 117 unused.
108 used; 118 unused.
109 used; 126 unused.
110 used; 125 unused.
111 used; 127 unused.
112 used; 127 unused.
113 used; 131 unused.
114 used; 134 unused.
115 used; 135 unused.
116 used; 134 unused.
117 used; 134 unused.
118 used; 133 unused.
```

```
119 used; 132 unused.
120 used; 132 unused.
121 used; 132 unused.
121 used; 131 unused.
122 used; 130 unused.
123 used; 129 unused.
124 used; 128 unused.
125 used; 127 unused.
126 used; 126 unused.
127 used; 125 unused.
128 used; 124 unused.
129 used; 123 unused.
130 used; 122 unused.
131 used; 121 unused.
132 used; 120 unused.
133 used; 120 unused.
134 used; 119 unused.
134 used; 118 unused.
135 used; 117 unused.
136 used; 116 unused.
137 used; 115 unused.
138 used; 115 unused.
139 used; 114 unused.
139 used; 113 unused.
140 used; 112 unused.
141 used; 111 unused.
142 used; 111 unused.
143 used; 110 unused.
144 used; 109 unused.
145 used; 108 unused.
146 used; 107 unused.
147 used; 107 unused.
148 used; 106 unused.
148 used; 106 unused.
149 used; 105 unused.
150 used; 104 unused.
151 used; 103 unused.
152 used; 103 unused.
153 used; 102 unused.
154 used; 102 unused.
154 used; 101 unused.
155 used; 100 unused.
156 used; 99 unused.
156 used; 99 unused.
156 used; 98 unused.
157 used; 97 unused.
158 used; 96 unused.
159 used; 96 unused.
160 used; 95 unused.
161 used; 94 unused.
162 used; 93 unused.
163 used; 92 unused.
164 used; 91 unused.
165 used; 90 unused.
166 used; 89 unused.
```

```
167 used; 88 unused.
167 used; 87 unused.
168 used; 86 unused.
169 used; 85 unused.
170 used; 84 unused.
171 used; 83 unused.
172 used; 84 unused.
173 used; 83 unused.
174 used; 82 unused.
175 used; 81 unused.
176 used; 80 unused.
177 used; 79 unused.
178 used; 79 unused.
178 used; 79 unused.
178 used; 78 unused.
178 used; 77 unused.
179 used; 76 unused.
179 used; 75 unused.
179 used; 74 unused.
180 used; 73 unused.
181 used; 72 unused.
182 used; 71 unused.
182 used; 70 unused.
182 used; 69 unused.
183 used; 68 unused.
184 used; 67 unused.
185 used; 67 unused.
186 used; 66 unused.
186 used; 65 unused.
187 used; 64 unused.
187 used; 63 unused.
187 used; 62 unused.
187 used; 61 unused.
188 used; 60 unused.
188 used; 59 unused.
189 used; 58 unused.
val gilmore_1 : bool list = [true]
- : dummy_interactive = END_INTERACTIVE
#
```

prolog.ml

```
#use "prolog.ml";;
```

```
# #use "prolog.ml";;
val renamerule :
  int ->
  fol formula list * fol formula ->
  (fol formula list * fol formula) * int = <fun>
val backchain :
  (fol formula list * fol formula) list ->
  int ->
  int -> (string, term) func -> fol formula list -> (string, term) func =
  <fun>
val hornify : 'a formula list -> 'a formula list * 'a formula = <fun>
val hornprove : fol formula -> (string, term) func * int = <fun>
- : dummy_interactive = START_INTERACTIVE
// prolog.p001
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
Searching with depth limit 8
val p32 : (string, term) func * int = (<func>, 8)
- : dummy_interactive = END_INTERACTIVE
val parserule : string -> fol formula list * fol formula = <fun>
val simpleprolog : string list -> string -> (string, term) func = <fun>
- : dummy_interactive = START_INTERACTIVE
val lerules : string list = ["0 <= X"; "S(X) <= S(Y) :- X <= Y"]
// prolog.p003
- : (string, term) func = <func>
val env : (string, term) func = <func>
// prolog.p005
- : term = <<|S(_1)|>>
- : dummy_interactive = END_INTERACTIVE
val prolog : string list -> string -> fol formula list = <fun>
- : dummy_interactive = START_INTERACTIVE
// prolog.p006
- : fol formula list = [<<X = S(S(_3))>>]
val appendrules : string list =
  ["append(nil,L,L)"; "append(H::T,L,H::A) :- append(T,L,A)"]
// prolog.p007
- : fol formula list = [<<Z = 1::2::3::4::nil>>]
// prolog.p008
- : fol formula list = [<<Y = 3::4::nil>>]
// prolog.p009
- : fol formula list = [<<X = 1::2::nil>>]
// prolog.p010
- : fol formula list = [<<X = nil>>; <<Y = 1::2::3::4::nil>>]
```

```
val sortrules : string list =
  ["sort(X,Y) :- perm(X,Y),sorted(Y)"; "sorted(nil)"; "sorted(X::nil)";
   "sorted(X::Y::Z) :- X <= Y, sorted(Y::Z)"; "perm(nil,nil)";
   "perm(X::Y,U::V) :- delete(U,X::Y,Z), perm(Z,V)";
   "delete(X,X::Y,Y)"; "delete(X,Y::Z,Y::W) :- delete(X,Z,W)";
   "0 <= X"; "S(X) <= S(Y) :- X <= Y"]
// prolog.p012
- : fol formula list =
[<<X = 0::S(0)::S(0)::S(S(0))::S(S(S(S(0))))::nil>>]
val badrules : string list =
  ["sort(X,Y) :- sorted(Y), perm(X,Y)"; "sorted(nil)";
   "sorted(X::nil)"; "sorted(X::Y::Z) :- X <= Y, sorted(Y::Z)";
   "perm(nil,nil)"; "perm(X::Y,U::V) :- delete(U,X::Y,Z), perm(Z,V)";
   "delete(X,X::Y,Y)"; "delete(X,Y::Z,Y::W) :- delete(X,Z,W)";
   "0 <= X"; "S(X) <= S(Y) :- X <= Y"]
- : dummy_interactive = END_INTERACTIVE
#
```

meson.ml

```
#use "meson.ml";;
```

```
# #use "meson.ml";;
- : dummy_interactive = START_INTERACTIVE
// meson.p001
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
- : int = 2
// meson.p002
Searching with depth limit 0
- : int = 0
// meson.p003
Searching with depth limit 0
- : int = 0
- : dummy_interactive = END_INTERACTIVE
val contrapositives :
  'a formula list -> ('a formula list * 'a formula) list = <fun>
val mexpand :
  (fol formula list * fol formula) list ->
  fol formula list ->
  fol formula ->
  ((string, term) func * int * int -> 'a) ->
  (string, term) func * int * int -> 'a = <fun>
File "/home/student/Harrison/OCaml/meson.ml", line 56, characters 5-56:
Warning S: this expression should have type unit.
val puremeson : fol formula -> int = <fun>
val meson : fol formula -> int list = <fun>
- : dummy_interactive = START_INTERACTIVE
// meson.p004
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
Searching with depth limit 8
val davis_putnam_example : int list = [8]
- : dummy_interactive = END_INTERACTIVE
val equal : (string, term) func -> fol formula -> fol formula -> bool =
  <fun>
val expand2 :
  ('a -> ('b * int * 'c -> 'd) -> 'b * int * 'c -> 'd) ->
  'a ->
  int -> 'a -> int -> int -> ('b * int * 'c -> 'd) -> 'b -> 'c -> 'd =
  <fun>
val mexpand :
  (fol formula list * fol formula) list ->
  fol formula list ->
  fol formula ->
  ((string, term) func * int * int -> 'a) ->
  (string, term) func * int * int -> 'a = <fun>
val mexpands :
  (fol formula list * fol formula) list ->
  fol formula list ->
  fol formula list ->
  ((string, term) func * int * int -> 'a) ->
  (string, term) func * int * int -> 'a = <fun>
File "/home/student/Harrison/OCaml/meson.ml", line 114, characters 5-56:
Warning S: this expression should have type unit.
val puremeson : fol formula -> int = <fun>
val meson : fol formula -> int list = <fun>
- : dummy_interactive = START_INTERACTIVE
// meson.p007
CPU time (user): 0.
val prop_1 : int list = []
// meson.p008
CPU time (user): 0.
val prop_2 : int list = []
// meson.p009
CPU time (user): 0.
val prop_3 : int list = []
// meson.p010
CPU time (user): 0.
val prop_4 : int list = []
// meson.p011
CPU time (user): 0.
val prop_5 : int list = []
// meson.p012
CPU time (user): 0.
val prop_6 : int list = []
// meson.p013
CPU time (user): 0.
val prop_7 : int list = []
// meson.p014
CPU time (user): 0.
val prop_8 : int list = []
// meson.p015
CPU time (user): 0.
val prop_9 : int list = []
// meson.p016
CPU time (user): 0.
val prop_10 : int list = []
// meson.p017
CPU time (user): 0.
val prop_11 : int list = []
// meson.p018
CPU time (user): 0.
val prop_12 : int list = []
// meson.p019
CPU time (user): 0.
val prop_13 : int list = []
// meson.p020
CPU time (user): 0.
```

```
val prop_14 : int list = []
// meson.p021
CPU time (user): 0.
val prop_15 : int list = []
// meson.p022
CPU time (user): 0.
val prop_16 : int list = []
// meson.p023
CPU time (user): 0.
val prop_17 : int list = []
// meson.p024
Searching with depth limit 0
Searching with depth limit 1
CPU time (user): 0.
val p18 : int list = [1]
// meson.p025
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
CPU time (user): 0.
val p19 : int list = [2]
// meson.p026
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
CPU time (user): 0.
val p20 : int list = [3]
// meson.p027
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
CPU time (user): 0.
val p21 : int list = [2; 3; 2]
// meson.p028
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
CPU time (user): 0.004
val p22 : int list = [2; 2]
// meson.p029
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
CPU time (user): 0.
val p23 : int list = [2; 1]
// meson.p030
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
CPU time (user): 0.004001
val p24 : int list = [4]
// meson.p031
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
CPU time (user): 0.
val p25 : int list = [2; 3]
// meson.p032
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 0
Searching with depth limit 1
CPU time (user): 0.008
val p26 : int list = [5; 5; 1; 1]
// meson.p033
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
CPU time (user): 0.004
val p27 : int list = [5]
// meson.p034
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
CPU time (user): 0.
val p28 : int list = [1; 2; 2; 2]
// meson.p035
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
CPU time (user): 0.
val p29 : int list = [3; 2; 2; 3]
// meson.p036
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
CPU time (user): 0.004
val p30 : int list = [4]
// meson.p037
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
CPU time (user): 0.004
val p31 : int list = [4]
// meson.p038
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
CPU time (user): 0.004
val p32 : int list = [7]
// meson.p039
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
CPU time (user): 0.004001
val p33 : int list = [3; 3; 3]
// meson.p040
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 0
Searching with depth limit 1
CPU time (user): 0.016001
val p34 : int list =
  [3; 3; 3; 1; 2; 2; 2; 1; 3; 2; 2; 1; 3; 2; 2; 1; 2; 1; 2; 1; 3; 1; 2;
   1; 2; 1; 2; 1; 1; 1; 1; 1]
// meson.p041
Searching with depth limit 0
Searching with depth limit 1
CPU time (user): 0.
val p35 : int list = [1]
```

```
// meson.p042
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
CPU time (user): 0.004
val p36 : int list = [3]
// meson.p043
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
CPU time (user): 0.004
val p37 : int list = [1; 3]
// meson.p044
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
Searching with depth limit 8
Searching with depth limit 9
Searching with depth limit 10
Searching with depth limit 11
Searching with depth limit 12
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
Searching with depth limit 8
Searching with depth limit 9
Searching with depth limit 10
Searching with depth limit 11
Searching with depth limit 12
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
Searching with depth limit 8
Searching with depth limit 9
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
Searching with depth limit 8
Searching with depth limit 9
CPU time (user): 0.456029
val p38 : int list = [12; 12; 9; 9]
// meson.p045
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
CPU time (user): 0.
val p39 : int list = [4]
// meson.p046
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
CPU time (user): 0.004
val p40 : int list = [6]
// meson.p047
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
CPU time (user): 0.00800099999999
val p41 : int list = [6]
// meson.p048
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
Searching with depth limit 8
Searching with depth limit 9
Searching with depth limit 10
Searching with depth limit 11
Searching with depth limit 12
CPU time (user): 0.396025
val p42 : int list = [12]
// meson.p049
```

```
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
Searching with depth limit 8
Searching with depth limit 9
Searching with depth limit 10
Searching with depth limit 11
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
Searching with depth limit 8
Searching with depth limit 9
Searching with depth limit 10
Searching with depth limit 11
CPU time (user): 0.48403
val p43 : int list = [11; 11]
// meson.p050
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
CPU time (user): 0.004
val p44 : int list = [6]
// meson.p051
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
Searching with depth limit 8
Searching with depth limit 9
Searching with depth limit 10
Searching with depth limit 11
Searching with depth limit 12
Searching with depth limit 13
Searching with depth limit 14
Searching with depth limit 15
Searching with depth limit 16
Searching with depth limit 17
Searching with depth limit 18
Searching with depth limit 19
Searching with depth limit 20
Searching with depth limit 21
Searching with depth limit 22
Searching with depth limit 23
Searching with depth limit 24
CPU time (user): 0.896056
val p45 : int list = [24]
// meson.p052
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
Searching with depth limit 8
Searching with depth limit 9
Searching with depth limit 10
Searching with depth limit 11
Searching with depth limit 12
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
CPU time (user): 0.476029
val p46 : int list = [12; 2]
// meson.p053
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
Searching with depth limit 8
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
CPU time (user): 0.036003
val p55 : int list = [8; 3]
// meson.p054
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
CPU time (user): 0.004
val p57 : int list = [3]
// meson.p055
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
CPU time (user): 0.
val p58 : int list = [3]
// meson.p056
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
CPU time (user): 0.004
val p59 : int list = [6]
// meson.p057
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
CPU time (user): 0.
val p60 : int list = [2; 3]
// meson.p060
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
CPU time (user): 0.00399999999999
val gilmore_3 : int list = [4]
// meson.p061
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
Searching with depth limit 8
CPU time (user): 0.00800100000001
val gilmore_4 : int list = [8]
// meson.p062
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
CPU time (user): 0.00399999999999
val gilmore_5 : int list = [4]
// meson.p063
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
```

```
CPU time (user): 0.
val gilmore_6 : int list = [2]
// meson.p064
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
Searching with depth limit 8
CPU time (user): 0.00800100000001
val gilmore_7 : int list = [8]
// meson.p065
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
CPU time (user): 0.004
val gilmore_8 : int list = [4]
// meson.p067
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
CPU time (user): 0.284018
val gilmore_9a : int list = [7]
// meson.p068
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
Searching with depth limit 8
CPU time (user): 0.00800000000001
val davis_putnam_example : int list = [8]
// meson.p069
- : int list = []
- : dummy_interactive = END_INTERACTIVE
#
```

skolems.ml

```
#use "skolems.ml";;
```

```
# #use "skolems.ml";;
val rename_term : term -> term = <fun>
val rename_form : fol formula -> fol formula = <fun>
val skolems :
  fol formula list -> string list -> fol formula list * string list =
  <fun>
val skolemizes : fol formula list -> fol formula list = <fun>
- : dummy_interactive = START_INTERACTIVE
// skolems.p001
- : fol formula list =
[<<old_+(c_x,c_y) = old_2>>; <<forall x. old_+(x,old_1) = f_y(x)>>]
- : dummy_interactive = END_INTERACTIVE
#
```

equal.ml

```
#use "equal.ml";;
```

```
# #use "equal.ml";;
val is_eq : fol formula -> bool = <fun>
val mk_eq : term -> term -> fol formula = <fun>
val dest_eq : fol formula -> term * term = <fun>
val lhs : fol formula -> term = <fun>
val rhs : fol formula -> term = <fun>
val predicates : fol formula -> (string * int) list = <fun>
val function_congruence : string * int -> fol formula list = <fun>
- : dummy_interactive = START_INTERACTIVE
// equal.p001
- : fol formula list =
[<<forall x1 x2 x3 y1 y2 y3.
     x1 = y1 /\ x2 = y2 /\ x3 = y3 ==> f(x1,x2,x3) = f(y1,y2,y3)>>]
// equal.p002
- : fol formula list =
[<<forall x1 x2 y1 y2. x1 = y1 /\ x2 = y2 ==> x1 + x2 = y1 + y2>>]
- : dummy_interactive = END_INTERACTIVE
val predicate_congruence : string * int -> fol formula list = <fun>
val equivalence_axioms : fol formula list =
  [<<forall x. x = x>>; <<forall x y z. x = y /\ x = z ==> y = z>>]
val equalitize : fol formula -> fol formula = <fun>
- : dummy_interactive = START_INTERACTIVE
val ewd : fol formula =
  <<(forall x. x = x) /\
    (forall x y z. x = y /\ x = z ==> y = z) /\
    (forall x1 y1. x1 = y1 ==> f(x1) ==> f(y1)) /\
    (forall x1 y1. x1 = y1 ==> g(x1) ==> g(y1)) ==>
    (forall x. f(x) ==> g(x)) /\
    (exists x. f(x)) /\ (forall x y. g(x) /\ g(y) ==> x = y) ==>
    (forall y. g(y) ==> f(y))>>
// equal.p003
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
- : int list = [6]
val wishnu : fol formula =
  <<(forall x. x = x) /\
    (forall x y z. x = y /\ x = z ==> y = z) /\
    (forall x1 y1. x1 = y1 ==> f(x1) = f(y1)) /\
    (forall x1 y1. x1 = y1 ==> g(x1) = g(y1)) ==>
    ((exists x. x = f(g(x)) /\ (forall x'. x' = f(g(x')) ==> x = x')) <=>
     (exists y. y = g(f(y)) /\ (forall y'. y' = g(f(y')) ==> y = y')))>>
// equal.p004
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
Searching with depth limit 8
Searching with depth limit 9
Searching with depth limit 10
Searching with depth limit 11
Searching with depth limit 12
Searching with depth limit 13
Searching with depth limit 14
Searching with depth limit 15
Searching with depth limit 16
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
Searching with depth limit 8
Searching with depth limit 9
Searching with depth limit 10
Searching with depth limit 11
Searching with depth limit 12
Searching with depth limit 13
Searching with depth limit 14
Searching with depth limit 15
Searching with depth limit 16
CPU time (user): 3.67223
- : int list = [16; 16]
- : dummy_interactive = END_INTERACTIVE
#
```

cong.ml

```
#use "cong.ml";;
```

```
# #use "cong.ml";;
val subterms : term -> term list = <fun>
val congruent : term partition -> term * term -> bool = <fun>
val emerge :
  term * term ->
  term partition * (term, term list) func ->
  term partition * (term, term list) func = <fun>
val predecessors :
  term -> (term, term list) func -> (term, term list) func = <fun>
val ccsatisfiable : fol formula list -> bool = <fun>
val ccvalid : fol formula -> bool = <fun>
- : dummy_interactive = START_INTERACTIVE
// cong.p001
- : bool = true
// cong.p002
- : bool = false
- : dummy_interactive = END_INTERACTIVE
#
```

rewrite.ml

```
#use "rewrite.ml";;
```

```
# #use "rewrite.ml";;
val rewrite1 : fol formula list -> term -> term = <fun>
val rewrite : fol formula list -> term -> term = <fun>
- : dummy_interactive = START_INTERACTIVE
// rewrite.p001
- : term = <<|S(S(S(S(S(S(S(S(S(S(0))))))))))|>>
- : dummy_interactive = END_INTERACTIVE
#
```

order.ml

```
#use "order.ml";;
```

```
# #use "order.ml";;
val termsize : term -> int = <fun>
- : dummy_interactive = START_INTERACTIVE
val s : term = <<|f(x,x,x)|>>
val t : term = <<|g(x,y)|>>
// order.p001
- : bool = true
val i : (string, term) func = <func>
// order.p002
- : bool = false
- : dummy_interactive = END_INTERACTIVE
val lexord : ('a -> 'a -> bool) -> 'a list -> 'a list -> bool = <fun>
val lpo_gt :
  (string * int -> string * int -> bool) -> term -> term -> bool =
  <fun>
val lpo_ge :
  (string * int -> string * int -> bool) -> term -> term -> bool =
  <fun>
val weight : 'a list -> 'a * 'b -> 'a * 'b -> bool = <fun>
#
```

completion.ml

```
#use "completion.ml";;
```

```
# #use "completion.ml";;
val renamepair : fol formula * fol formula -> fol formula * fol formula =
  <fun>
val listcases :
  ('a -> ('b -> 'a -> 'c) -> 'd list) ->
  ('b -> 'a list -> 'c) -> 'a list -> 'd list -> 'd list = <fun>
val overlaps :
  term * term -> term -> ((string, term) func -> term -> 'a) -> 'a list =
  <fun>
File "/home/student/Harrison/OCaml/completion.ml", line 35, characters 33-113:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
(Exists (_, _)|Forall (_, _)|Iff (_, _)|Imp (_, _)|Or (_, _)|And (_, _)|
Not _|True|False)
File "/home/student/Harrison/OCaml/completion.ml", line 35, characters 10-113:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
(Exists (_, _)|Forall (_, _)|Iff (_, _)|Imp (_, _)|Or (_, _)|And (_, _)|
Not _|True|False)
val crit1 : fol formula -> fol formula -> fol formula list = <fun>
val critical_pairs : fol formula -> fol formula -> fol formula list =
  <fun>
- : dummy_interactive = START_INTERACTIVE
// completion.p001
- : fol formula list = [<<f(g(x0)) = g(f(x0))>>; <<g(x1) = g(x1)>>]
- : dummy_interactive = END_INTERACTIVE
File "/home/student/Harrison/OCaml/completion.ml", line 55, characters 33-205:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
(Exists (_, _)|Forall (_, _)|Iff (_, _)|Imp (_, _)|Or (_, _)|And (_, _)|
Not _|True|False)
val normalize_and_orient :
  (term -> term -> bool) ->
  fol formula list -> fol formula -> term * term = <fun>
val status : 'a list * 'b list * 'c list -> 'a list -> unit = <fun>
val complete :
  (term -> term -> bool) ->
  fol formula list * fol formula list * fol formula list ->
  fol formula list = <fun>
- : dummy_interactive = START_INTERACTIVE
val eqs : fol formula list =
  [<<1 * x = x>>; <<i(x) * x = 1>>; <<(x * y) * z = x * y * z>>]
val ord : term -> term -> bool = <fun>
4 equations and 8 pending critical pairs + 0 deferred
5 equations and 12 pending critical pairs + 0 deferred
6 equations and 16 pending critical pairs + 0 deferred
7 equations and 27 pending critical pairs + 0 deferred
```

```
8 equations and 51 pending critical pairs + 0 deferred
9 equations and 70 pending critical pairs + 0 deferred
10 equations and 81 pending critical pairs + 0 deferred
11 equations and 78 pending critical pairs + 0 deferred
12 equations and 85 pending critical pairs + 0 deferred
13 equations and 114 pending critical pairs + 0 deferred
14 equations and 151 pending critical pairs + 0 deferred
15 equations and 180 pending critical pairs + 0 deferred
16 equations and 247 pending critical pairs + 0 deferred
17 equations and 298 pending critical pairs + 0 deferred
18 equations and 356 pending critical pairs + 0 deferred
19 equations and 404 pending critical pairs + 0 deferred
20 equations and 485 pending critical pairs + 0 deferred
21 equations and 530 pending critical pairs + 0 deferred
22 equations and 583 pending critical pairs + 0 deferred
23 equations and 642 pending critical pairs + 0 deferred
24 equations and 730 pending critical pairs + 0 deferred
25 equations and 779 pending critical pairs + 0 deferred
26 equations and 794 pending critical pairs + 0 deferred
27 equations and 819 pending critical pairs + 1 deferred
28 equations and 918 pending critical pairs + 1 deferred
29 equations and 901 pending critical pairs + 1 deferred
30 equations and 1005 pending critical pairs + 1 deferred
31 equations and 1086 pending critical pairs + 1 deferred
32 equations and 1155 pending critical pairs + 1 deferred
32 equations and 1000 pending critical pairs + 1 deferred
32 equations and 0 pending critical pairs + 1 deferred
32 equations and 0 pending critical pairs + 0 deferred
val eqs' : fol formula list =
  [<<i(x4 * x5) = i(x5) * i(x4)>>; <<x1 * i(x5 * x1) = i(x5)>>;
   <<i(x4) * x1 * i(x3 * x1) = i(x4) * i(x3)>>;
   <<x1 * i(i(x4) * i(x3) * x1) = x3 * x4>>;
   <<i(x3 * x5) * x0 = i(x5) * i(x3) * x0>>;
   <<i(x4 * x5 * x6 * x3) * x0 = i(x3) * i(x4 * x5 * x6) * x0>>;
   <<i(x0 * i(x1)) = x1 * i(x0)>>; <<i(i(x2 * x1) * x2) = x1>>;
   <<i(i(x4) * x2) * x0 = i(x2) * x4 * x0>>;
   <<x1 * i(x2 * x1) * x2 = 1>>;
   <<x1 * i(i(x4 * x5) * x1) * x3 = x4 * x5 * x3>>;
   <<i(x3 * i(x1 * x2)) = x1 * x2 * i(x3)>>;
   <<i(i(x3 * i(x1 * x2)) * i(x5 * x6)) * x1 * x2 * x0 = x5 * x6 * x3 *
     x0>>;
   <<x1 * x2 * i(x1 * x2) = 1>>; <<x2 * x3 * i(x2 * x3) * x1 = x1>>;
   <<i(x3 * x4) * x3 * x1 = i(x4) * x1>>;
   <<i(x1 * x3 * x4) * x1 * x3 * x4 * x0 = x0>>;
   <<i(x1 * i(x3)) * x1 * x4 = x3 * x4>>;
   <<i(i(x5 * x2) * x5) * x0 = x2 * x0>>;
   <<i(x4 * i(x1 * x2)) * x4 * x0 = x1 * x2 * x0>>; <<i(i(x1)) = x1>>;
   <<i(1) = 1>>; <<x0 * i(x0) = 1>>; <<x0 * i(x0) * x3 = x3>>;
   <<i(x2 * x3) * x2 * x3 * x1 = x1>>; <<x1 * 1 = x1>>;
   <<i(1) * x1 = x1>>; <<i(i(x0)) * x1 = x0 * x1>>;
   <<i(x1) * x1 * x2 = x2>>; <<1 * x = x>>; <<i(x) * x = 1>>;
   <<(x * y) * z = x * y * z>>]
// completion.p002
- : term = <<|z|>>
- : dummy_interactive = END_INTERACTIVE
```

```
File "/home/student/Harrison/OCaml/completion.ml", line 112, characters 2-221:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
(Exists (_, _)|Forall (_, _)|Iff (_, _)|Imp (_, _)|Or (_, _)|And (_, _)|
Not _|True|False)::_
val interreduce :
  fol formula list -> fol formula list -> fol formula list = <fun>
- : dummy_interactive = START_INTERACTIVE
// completion.p003
- : fol formula list =
[<<i(x4 * x5) = i(x5) * i(x4)>>; <<i(i(x1)) = x1>>; <<i(1) = 1>>;
 <<x0 * i(x0) = 1>>; <<x0 * i(x0) * x3 = x3>>; <<x1 * 1 = x1>>;
 <<i(x1) * x1 * x2 = x2>>; <<1 * x = x>>; <<i(x) * x = 1>>;
 <<(x * y) * z = x * y * z>>]
- : dummy_interactive = END_INTERACTIVE
val complete_and_simplify :
  string list -> fol formula list -> fol formula list = <fun>
- : dummy_interactive = START_INTERACTIVE
// completion.p004
2 equations and 4 pending critical pairs + 0 deferred
3 equations and 9 pending critical pairs + 0 deferred
3 equations and 0 pending critical pairs + 0 deferred
- : fol formula list =
[<<x0 * i(x0) * x3 = x3>>; <<i(i(x0)) * x1 = x0 * x1>>;
 <<i(a) * a * b = b>>]
// completion.p005
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
- : int list = [5; 4]
// completion.p006
- : fol formula = <<~z = x * y \/ f_w(x,z) = y>>
- : dummy_interactive = END_INTERACTIVE
- : dummy_interactive = START_INTERACTIVE
// completion.p008
val eqs : fol formula list = [<<(a * b) * b * c = b>>]
2 equations and 8 pending critical pairs + 0 deferred
3 equations and 18 pending critical pairs + 0 deferred
3 equations and 0 pending critical pairs + 0 deferred
- : fol formula list =
[<<(x3 * x0 * x1) * x1 = x0 * x1>>; <<x1 * (x1 * x2) * x5 = x1 * x2>>;
 <<(a * b) * b * c = b>>]
// completion.p010
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
- : int list = [5; 4]
// completion.p011
- : fol formula = <<~z = x * y \/ f_w(x,z) = y>>
val eqs : fol formula list =
  [<<f(a,a * b) = b>>; <<g(a * b,b) = a>>; <<1 * a = a>>;
   <<a * 1 = a>>]
// completion.p012
5 equations and 8 pending critical pairs + 0 deferred
6 equations and 10 pending critical pairs + 0 deferred
7 equations and 11 pending critical pairs + 0 deferred
8 equations and 12 pending critical pairs + 0 deferred
8 equations and 0 pending critical pairs + 0 deferred
- : fol formula list =
[<<g(x1,x1) = 1>>; <<g(x0,1) = x0>>; <<f(1,x1) = x1>>;
 <<f(x0,x0) = 1>>; <<f(a,a * b) = b>>; <<g(a * b,b) = a>>;
 <<1 * a = a>>; <<a * 1 = a>>]
val eqs : fol formula list = [<<f(a,f(b,c,a),d) = c>>]
val eqs : fol formula list =
  [<<f(a,f(b,c,a),d) = c>>; <<f(a,b,c) = g(a,b)>>; <<g(a,b) = h(b)>>]
// completion.p014
4 equations and 11 pending critical pairs + 0 deferred
4 equations and 0 pending critical pairs + 0 deferred
- : fol formula list =
[<<h(h(x2)) = x2>>; <<f(a,b,c) = h(b)>>; <<g(a,b) = h(b)>>]
- : dummy_interactive = END_INTERACTIVE
#
```

eqelim.ml

```
#use "eqelim.ml";;
```

```
# #use "eqelim.ml";;
- : dummy_interactive = START_INTERACTIVE
// eqelim.p001
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
Searching with depth limit 8
Searching with depth limit 9
Searching with depth limit 10
Searching with depth limit 11
Searching with depth limit 12
Searching with depth limit 13
- : int list = [13]
// eqelim.p002
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
Searching with depth limit 8
Searching with depth limit 9
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
- : int list = [4; 3; 9; 3; 2; 7]
- : dummy_interactive = END_INTERACTIVE
val modify_S : fol formula list -> fol formula list list = <fun>
val modify_T : fol formula list -> fol formula list = <fun>
val is_nonvar : term -> bool = <fun>
val find_nestnonvar : term -> term = <fun>
File "/home/student/Harrison/OCaml/eqelim.ml", line 64, characters 2-144:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
(Exists (_, _)|Forall (_, _)|Iff (_, _)|Imp (_, _)|Or (_, _)|And (_, _)|True|
False)
val find_nvsubterm : fol formula -> term = <fun>
val replacet : (term, term) func -> term -> term = <fun>
val replace : (term, term) func -> fol formula -> fol formula = <fun>
val emodify : string list -> fol formula list -> fol formula list =
  <fun>
val modify_E : fol formula list -> fol formula list = <fun>
val brand : fol formula list list -> fol formula list list = <fun>
File "/home/student/Harrison/OCaml/eqelim.ml", line 111, characters 5-56:
Warning S: this expression should have type unit.
val bpuremeson : fol formula -> int = <fun>
val bmeson : fol formula -> int list = <fun>
- : dummy_interactive = START_INTERACTIVE
// eqelim.p003
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
Searching with depth limit 8
Searching with depth limit 9
Searching with depth limit 10
Searching with depth limit 11
Searching with depth limit 12
Searching with depth limit 13
Searching with depth limit 14
Searching with depth limit 15
Searching with depth limit 16
Searching with depth limit 17
Searching with depth limit 18
Searching with depth limit 19
Searching with depth limit 20
Searching with depth limit 21
Searching with depth limit 22
Searching with depth limit 23
Searching with depth limit 24
Searching with depth limit 25
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
Searching with depth limit 8
Searching with depth limit 9
Searching with depth limit 10
Searching with depth limit 11
Searching with depth limit 12
Searching with depth limit 13
Searching with depth limit 14
Searching with depth limit 15
Searching with depth limit 16
Searching with depth limit 17
Searching with depth limit 18
Searching with depth limit 19
Searching with depth limit 20
Searching with depth limit 21
Searching with depth limit 22
Searching with depth limit 23
Searching with depth limit 24
Searching with depth limit 25
CPU time (user): 44.8028
- : int list = [25; 25]
// eqelim.p004
Note: emson source code commented out because it causes an error
// eqelim.p005
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
Searching with depth limit 8
Searching with depth limit 9
Searching with depth limit 10
Searching with depth limit 11
Searching with depth limit 12
Searching with depth limit 13
Searching with depth limit 14
Searching with depth limit 15
Searching with depth limit 16
Searching with depth limit 17
Searching with depth limit 18
Searching with depth limit 19
CPU time (user): 295.410462
- : int list = [19]
- : dummy_interactive = END_INTERACTIVE
- : dummy_interactive = START_INTERACTIVE
val emeson : fol formula -> int list = <fun>
val ewd : fol formula =
  <<(forall x. f(x) ==> g(x)) /\
    (exists x. f(x)) /\ (forall x y. g(x) /\ g(y) ==> x = y) ==>
    (forall y. g(y) ==> f(y))>>
val wishnu : fol formula =
  <<(exists x. x = f(g(x)) /\ (forall x'. x' = f(g(x')) ==> x = x')) <=>
    (exists y. y = g(f(y)) /\ (forall y'. y' = g(f(y')) ==> y = y'))>>
val group1 : fol formula =
  <<(forall x y z. x * y * z = (x * y) * z) /\
    (forall x. e * x = x) /\ (forall x. i(x) * x = e) ==>
    (forall x. x * e = x)>>
val group2 : fol formula =
  <<(forall x y z. x * y * z = (x * y) * z) /\
    (forall x. e * x = x) /\ (forall x. i(x) * x = e) ==>
    (forall x. x * i(x) = e)>>
// eqelim.p006
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
Searching with depth limit 8
Searching with depth limit 9
CPU time (user): 0.024002
- : int list = [9]
// eqelim.p007
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
CPU time (user): 0.00400000000002
- : int list = [6]
- : dummy_interactive = END_INTERACTIVE
#
```

paramodulation.ml

```
#use "paramodulation.ml";;
```

```
# #use "paramodulation.ml";;
val overlapl :
  term * term ->
  fol formula -> ((string, term) func -> fol formula -> 'a) -> 'a list =
  <fun>
val overlapc :
  term * term ->
  fol formula list ->
  ((string, term) func -> fol formula list -> 'a) -> 'a list -> 'a list =
  <fun>
val paramodulate :
  fol formula list -> fol formula list -> fol formula list list = <fun>
val para_clauses :
  fol formula list -> fol formula list -> fol formula list list = <fun>
val paraloop : fol formula list list * fol formula list list -> bool =
  <fun>
val pure_paramodulation : fol formula -> bool = <fun>
val paramodulation : fol formula -> bool list = <fun>
- : dummy_interactive = START_INTERACTIVE
// paramodulation.p001
0 used; 4 unused.
1 used; 3 unused.
2 used; 4 unused.
3 used; 8 unused.
4 used; 11 unused.
5 used; 13 unused.
6 used; 21 unused.
7 used; 40 unused.
8 used; 64 unused.
9 used; 83 unused.
10 used; 108 unused.
11 used; 125 unused.
- : bool list = [true]
- : dummy_interactive = END_INTERACTIVE
#
```

decidable.ml

```
#use decidable ml";;
```

```
# #use "decidable.ml";;
- : dummy_interactive = START_INTERACTIVE
val los : fol formula =
  <<(forall x y z. P(x,y) /\ P(y,z) ==> P(x,z)) /\
    (forall x y z. Q(x,y) /\ Q(y,z) ==> Q(x,z)) /\
    (forall x y. P(x,y) ==> P(y,x)) /\ (forall x y. P(x,y) \/ Q(x,y)) ==>
    (forall x y. P(x,y)) \/ (forall x y. Q(x,y))>>
// decidable.p004
- : fol formula =
<<(((~P(x,y) \/ ~P(y,z)) \/ P(x,z)) /\
   ((~Q(x,y) \/ ~Q(y,z)) \/ Q(x,z)) /\
   (~P(x,y) \/ P(y,x)) /\ (P(x,y) \/ Q(x,y))) /\
  ~P(c_x,c_y) /\ ~Q(c_x',c_y')>>
// decidable.p005
0 ground instances tried; 0 items in list
0 ground instances tried; 0 items in list
1 ground instances tried; 5 items in list
2 ground instances tried; 7 items in list
3 ground instances tried; 9 items in list
4 ground instances tried; 11 items in list
5 ground instances tried; 15 items in list
6 ground instances tried; 17 items in list
7 ground instances tried; 19 items in list
8 ground instances tried; 21 items in list
9 ground instances tried; 25 items in list
10 ground instances tried; 27 items in list
11 ground instances tried; 29 items in list
12 ground instances tried; 31 items in list
13 ground instances tried; 35 items in list
14 ground instances tried; 37 items in list
15 ground instances tried; 39 items in list
16 ground instances tried; 41 items in list
17 ground instances tried; 45 items in list
18 ground instances tried; 47 items in list
19 ground instances tried; 49 items in list
20 ground instances tried; 51 items in list
21 ground instances tried; 55 items in list
22 ground instances tried; 56 items in list
23 ground instances tried; 58 items in list
24 ground instances tried; 60 items in list
25 ground instances tried; 64 items in list
26 ground instances tried; 66 items in list
27 ground instances tried; 68 items in list
28 ground instances tried; 70 items in list
29 ground instances tried; 74 items in list
30 ground instances tried; 76 items in list
31 ground instances tried; 78 items in list
32 ground instances tried; 80 items in list
33 ground instances tried; 84 items in list
```

```
34 ground instances tried; 86 items in list
35 ground instances tried; 88 items in list
36 ground instances tried; 90 items in list
37 ground instances tried; 94 items in list
38 ground instances tried; 96 items in list
39 ground instances tried; 98 items in list
40 ground instances tried; 100 items in list
41 ground instances tried; 104 items in list
42 ground instances tried; 106 items in list
43 ground instances tried; 107 items in list
44 ground instances tried; 109 items in list
- : int = 45
- : dummy_interactive = END_INTERACTIVE
val aedecide : fol formula -> bool = <fun>
- : dummy_interactive = START_INTERACTIVE
// decidable.p006
- : bool = true
- : dummy_interactive = END_INTERACTIVE
- : dummy_interactive = START_INTERACTIVE
// decidable.p007
val fm : fol formula = <<(forall x. p(x)) \/ (exists y. p(y))>>
- : fol formula = <<forall x. exists y. p(x) \/ p(y)>>
// decidable.p008
- : bool = true
// decidable.p009
- : bool = false
// decidable.p010
- : bool = true
- : dummy_interactive = END_INTERACTIVE
val separate : string -> fol formula list -> fol formula = <fun>
val pushquant : string -> fol formula -> fol formula = <fun>
val miniscope : fol formula -> fol formula = <fun>
- : dummy_interactive = START_INTERACTIVE
// decidable.p013
- : fol formula = <<(exists y. ~P(y)) \/ (forall x. P(x))>>
// decidable.p014
val fm : fol formula =
  <<((exists x. P(x)) /\
     (forall z. ~R(z)) /\ (exists w. ~U(w)) /\ (exists y. Q(y)) \/
     (exists x. P(x)) /\ (forall z. ~R(z)) /\ (exists y. Q(y)) \/
     (exists x. P(x)) /\ (exists w. ~U(w)) /\ (exists y. Q(y))) \/
    ~((exists x. P(x)) /\ (exists y. Q(y))) \/ (exists z. R(z))>>
// decidable.p015
- : fol formula =
<<forall z z' x y.
    exists x' w y'.
      (P(x') /\ ~R(z) /\ ~U(w) /\ Q(y') \/
       P(x') /\ ~R(z') /\ Q(w) \/ P(x') /\ ~U(w) /\ Q(y')) \/
      (~P(x) \/ ~Q(y)) \/ R(x')>>
- : dummy_interactive = END_INTERACTIVE
val wang : fol formula -> bool = <fun>
- : dummy_interactive = START_INTERACTIVE
// decidable.p016
- : bool = true
// decidable.p017
```

```
- : fol formula =
<<forall y y' y'' y''' y'''' y''''' y'''''' x y''''''' y'''''''' y''''''''' y''''''''''
y''''''''''' y'''''''''''' y''''''''''''' x' x''.
    exists x''' x'''' y'''''''''''''' x''''' y''''''''''''''' x'''''' x'''''''
y'''''''''''''''' x'''''''' y'''''''''''''''''.
      (((P(x''') /\ ~P(x''')) /\ P(y) /\ (~P(y) \/ P(y)) \/
        (P(x''') /\ ~P(x''')) /\ ~P(y') /\ (~P(y') \/ P(y')) \/
        (P(x''') /\ ~P(x''')) /\ (~P(y'') \/ P(y'')) \/
        P(x''') /\ P(y''') /\ ~P(y''') /\ (~P(y''') \/ P(y''')) \/
        P(x''') /\ P(y'''') /\ (~P(y'''') \/ P(y'''')) \/
        ~P(x''') /\
        P(y''''') /\ ~P(y''''') /\ (~P(y''''') \/ P(y''''')) \/
        ~P(x''') /\ ~P(y'''''') /\ (~P(y'''''') \/ P(y''''''))) /\
       (Q(x'''') /\ Q(y) \/ ~Q(y') /\ ~Q(x'''')) \/
       ((~P(x) \/ P(x)) /\
        (~P(x) \/ ~P(x''')) /\
        (P(x) \/ P(x'''')) /\
        (~P(y'''''''''''''') \/ P(y''''''''''''''))) /\
       (Q(x''''') /\ ~Q(y''''''''''''''') \/ ~Q(x) /\ Q(x))) /\
      (((Q(x'''''') /\ ~Q(x'''''')) /\ Q(y) /\ (~Q(y) \/ Q(y)) \/
        (Q(x'''''') /\ ~Q(x'''''')) /\ ~Q(y') /\ (~Q(y') \/ Q(y')) \/
        (Q(x'''''') /\ ~Q(x'''''')) /\ (~Q(y'') \/ Q(y'')) \/
        Q(x'''''') /\ Q(y''') /\ ~Q(y''') /\ (~Q(y''') \/ Q(y''')) \/
        Q(x'''''') /\ Q(y'''') /\ (~Q(y'''') \/ Q(y'''')) \/
        ~Q(x'''''') /\
        Q(y''''') /\ ~Q(y''''') /\ (~Q(y''''') \/ Q(y''''')) \/
        ~Q(x'''''') /\ ~Q(y'''''') /\ (~Q(y'''''') \/ Q(y''''''))) /\
       (P(x''''''') /\ P(y) \/ ~P(y') /\ ~P(x''''''')) \/
       ((~Q(x) \/ Q(x)) /\
        (~Q(x) \/ ~Q(x'''''')) /\
        (Q(x) \/ Q(x''''''')) /\
        (~Q(y'''''''''''''''') \/ Q(y''''''''''''''''))) /\
       (P(x'''''''') /\ ~P(y''''''''''''''''') \/ ~P(x) /\ P(x))) \/
      (((P(x''') /\ ~P(x''')) /\
        P(y''''''') /\ (~P(y''''''') \/ P(y''''''')) \/
        (P(x''') /\ ~P(x''')) /\
        ~P(y'''''''') /\ (~P(y'''''''') \/ P(y'''''''')) \/
        (P(x''') /\ ~P(x''')) /\ (~P(y''''''''') \/ P(y''''''''')) \/
        P(x''') /\
        P(y'''''''''') /\
        ~P(y'''''''''') /\ (~P(y'''''''''') \/ P(y'''''''''')) \/
        P(x''') /\
        P(y''''''''''') /\ (~P(y''''''''''') \/ P(y''''''''''')) \/
        ~P(x''') /\
        P(y'''''''''''') /\
        ~P(y'''''''''''') /\ (~P(y'''''''''''') \/ P(y'''''''''''')) \/
        ~P(x''') /\
        ~P(y''''''''''''') /\ (~P(y''''''''''''') \/ P(y'''''''''''''))) /\
       (Q(x'''') /\ ~Q(y'''''''''''''') \/ ~Q(y''''''') /\ Q(y''''''')) \/
       ((~P(x') \/ P(x')) /\
        (~P(x') \/ ~P(x''')) /\
        (P(x') \/ P(x'''')) /\
        (~P(y'''''''''''''') \/ P(y''''''''''''''))) /\
       (Q(x''''') /\ Q(x') \/ ~Q(x'') /\ ~Q(x'''''))) /\
      (((Q(y''''''''''''''') /\ ~Q(y''''''''''''''')) /\
        Q(y''''''') /\ (~Q(y''''''') \/ Q(y'''''''))) \/
        (Q(y''''''''''''''') /\ ~Q(y''''''''''''''')) /\
        ~Q(y'''''''') /\ (~Q(y'''''''') \/ Q(y'''''''')) \/
        (Q(y''''''''''''''') /\ ~Q(y''''''''''''''')) /\
        (~Q(y''''''''') \/ Q(y''''''''')) \/
        Q(y''''''''''''''') /\
        Q(y'''''''''') /\
        ~Q(y'''''''''') /\ (~Q(y'''''''''') \/ Q(y'''''''''')) \/
        Q(y''''''''''''''') /\
        Q(y''''''''''') /\ (~Q(y''''''''''') \/ Q(y''''''''''')) \/
        ~Q(y''''''''''''''') /\
        Q(y'''''''''''') /\
        ~Q(y'''''''''''') /\ (~Q(y'''''''''''') \/ Q(y'''''''''''')) \/
        ~Q(y''''''''''''''') /\
        ~Q(y''''''''''''') /\ (~Q(y''''''''''''') \/ Q(y'''''''''''''))) /\
       (P(x'''''') /\ ~P(x''''''') \/ ~P(y''''''') /\ P(y''''''')) \/
       ((~Q(x') \/ Q(x')) /\
        (~Q(x') \/ ~Q(y''''''''''''''')) /\
        (Q(x') \/ Q(x''''''')) /\ (~Q(x''''''') \/ Q(x'''''''))) /\
       (P(y'''''''''''''''') /\ P(x') \/
        ~P(x'') /\ ~P(y''''''''''''''''))))>>
- : dummy_interactive = END_INTERACTIVE
val atom : string -> string -> fol formula = <fun>
val premiss_A : string * string -> fol formula = <fun>
val premiss_E : string * string -> fol formula = <fun>
val premiss_I : string * string -> fol formula = <fun>
val premiss_O : string * string -> fol formula = <fun>
File "/home/student/Harrison/OCaml/decidable.ml", line 188, characters 2-309:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
(Iff (_, _)|Imp (_, _)|Or (_, _)|And (_, _)|Not _|Atom _|True|False)
val anglicize_premiss : fol formula -> string = <fun>
File "/home/student/Harrison/OCaml/decidable.ml", line 195, characters 24-147:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
(Exists (_, _)|Forall (_, _)|Iff (_, _)|Or (_, _)|And (_, _)|Not _|Atom _|
True|False)
val anglicize_syllogism : fol formula -> string = <fun>
val all_possible_syllogisms : fol formula list =
  [<<(forall x. M(x) ==> P(x)) /\ (forall x. S(x) ==> M(x)) ==>
     (forall x. S(x) ==> P(x))>>;
   <<(forall x. M(x) ==> P(x)) /\ (forall x. S(x) ==> M(x)) ==>
     (forall x. S(x) ==> ~P(x))>>;
   <<(forall x. M(x) ==> P(x)) /\ (forall x. S(x) ==> M(x)) ==>
     (exists x. S(x) /\ P(x))>>;
   <<(forall x. M(x) ==> P(x)) /\ (forall x. S(x) ==> M(x)) ==>
     (exists x. S(x) /\ ~P(x))>>;
   <<(forall x. M(x) ==> P(x)) /\ (forall x. M(x) ==> S(x)) ==>
     (forall x. S(x) ==> P(x))>>;
   <<(forall x. M(x) ==> P(x)) /\ (forall x. M(x) ==> S(x)) ==>
     (forall x. S(x) ==> ~P(x))>>;
   <<(forall x. M(x) ==> P(x)) /\ (forall x. M(x) ==> S(x)) ==>
     (exists x. S(x) /\ P(x))>>;
   <<(forall x. M(x) ==> P(x)) /\ (forall x. M(x) ==> S(x)) ==>
     (exists x. S(x) /\ ~P(x))>>;
   <<(forall x. M(x) ==> P(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
     (forall x. S(x) ==> P(x))>>;
   <<(forall x. M(x) ==> P(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
     (forall x. S(x) ==> ~P(x))>>;
   <<(forall x. M(x) ==> P(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
     (exists x. S(x) /\ P(x))>>;
   <<(forall x. M(x) ==> P(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
     (exists x. S(x) /\ ~P(x))>>;
   <<(forall x. M(x) ==> P(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
     (forall x. S(x) ==> P(x))>>;
   <<(forall x. M(x) ==> P(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
     (forall x. S(x) ==> ~P(x))>>;
   <<(forall x. M(x) ==> P(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
     (exists x. S(x) /\ P(x))>>;
   <<(forall x. M(x) ==> P(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
     (exists x. S(x) /\ ~P(x))>>;
   <<(forall x. M(x) ==> P(x)) /\ (exists x. S(x) /\ M(x)) ==>
     (forall x. S(x) ==> P(x))>>;
   <<(forall x. M(x) ==> P(x)) /\ (exists x. S(x) /\ M(x)) ==>
     (forall x. S(x) ==> ~P(x))>>;
   <<(forall x. M(x) ==> P(x)) /\ (exists x. S(x) /\ M(x)) ==>
     (exists x. S(x) /\ P(x))>>;
   <<(forall x. M(x) ==> P(x)) /\ (exists x. S(x) /\ M(x)) ==>
     (exists x. S(x) /\ ~P(x))>>;
   <<(forall x. M(x) ==> P(x)) /\ (exists x. M(x) /\ S(x)) ==>
     (forall x. S(x) ==> P(x))>>;
   <<(forall x. M(x) ==> P(x)) /\ (exists x. M(x) /\ S(x)) ==>
     (forall x. S(x) ==> ~P(x))>>;
   <<(forall x. M(x) ==> P(x)) /\ (exists x. M(x) /\ S(x)) ==>
     (exists x. S(x) /\ P(x))>>;
   <<(forall x. M(x) ==> P(x)) /\ (exists x. M(x) /\ S(x)) ==>
     (exists x. S(x) /\ ~P(x))>>;
   <<(forall x. M(x) ==> P(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
     (forall x. S(x) ==> P(x))>>;
   <<(forall x. M(x) ==> P(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
     (forall x. S(x) ==> ~P(x))>>;
   <<(forall x. M(x) ==> P(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
     (exists x. S(x) /\ P(x))>>;
   <<(forall x. M(x) ==> P(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
     (exists x. S(x) /\ ~P(x))>>;
   <<(forall x. M(x) ==> P(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
     (forall x. S(x) ==> P(x))>>;
   <<(forall x. M(x) ==> P(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
     (forall x. S(x) ==> ~P(x))>>;
   <<(forall x. M(x) ==> P(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
     (exists x. S(x) /\ P(x))>>;
   <<(forall x. M(x) ==> P(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
     (exists x. S(x) /\ ~P(x))>>;
   <<(forall x. P(x) ==> M(x)) /\ (forall x. S(x) ==> M(x)) ==>
     (forall x. S(x) ==> P(x))>>;
   <<(forall x. P(x) ==> M(x)) /\ (forall x. S(x) ==> M(x)) ==>
     (forall x. S(x) ==> ~P(x))>>;
   <<(forall x. P(x) ==> M(x)) /\ (forall x. S(x) ==> M(x)) ==>
     (exists x. S(x) /\ P(x))>>;
   <<(forall x. P(x) ==> M(x)) /\ (forall x. S(x) ==> M(x)) ==>
```

```
    (exists x. S(x) /\ ~P(x))>>;
  <<(forall x. P(x) ==> M(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(forall x. P(x) ==> M(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(forall x. P(x) ==> M(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(forall x. P(x) ==> M(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(forall x. P(x) ==> M(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(forall x. P(x) ==> M(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(forall x. P(x) ==> M(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(forall x. P(x) ==> M(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(forall x. P(x) ==> M(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(forall x. P(x) ==> M(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(forall x. P(x) ==> M(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(forall x. P(x) ==> M(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(forall x. P(x) ==> M(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(forall x. P(x) ==> M(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(forall x. P(x) ==> M(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(forall x. P(x) ==> M(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(forall x. P(x) ==> M(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(forall x. P(x) ==> M(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(forall x. P(x) ==> M(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(forall x. P(x) ==> M(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(forall x. P(x) ==> M(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(forall x. P(x) ==> M(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(forall x. P(x) ==> M(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(forall x. P(x) ==> M(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(forall x. P(x) ==> M(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(forall x. P(x) ==> M(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(forall x. P(x) ==> M(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(forall x. P(x) ==> M(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(forall x. M(x) ==> ~P(x)) /\ (forall x. S(x) ==> M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(forall x. M(x) ==> ~P(x)) /\ (forall x. S(x) ==> M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(forall x. M(x) ==> ~P(x)) /\ (forall x. S(x) ==> M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(forall x. M(x) ==> ~P(x)) /\ (forall x. S(x) ==> M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(forall x. M(x) ==> ~P(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(forall x. M(x) ==> ~P(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(forall x. M(x) ==> ~P(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(forall x. M(x) ==> ~P(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(forall x. M(x) ==> ~P(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(forall x. M(x) ==> ~P(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(forall x. M(x) ==> ~P(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(forall x. M(x) ==> ~P(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(forall x. M(x) ==> ~P(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(forall x. M(x) ==> ~P(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(forall x. M(x) ==> ~P(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(forall x. M(x) ==> ~P(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(forall x. M(x) ==> ~P(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(forall x. M(x) ==> ~P(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(forall x. M(x) ==> ~P(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(forall x. M(x) ==> ~P(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(forall x. M(x) ==> ~P(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(forall x. M(x) ==> ~P(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(forall x. M(x) ==> ~P(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(forall x. M(x) ==> ~P(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(forall x. M(x) ==> ~P(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(forall x. M(x) ==> ~P(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(forall x. M(x) ==> ~P(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
```

```
     (exists x. S(x) /\ P(x))>>;
   <<(forall x. M(x) ==> ~P(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
     (exists x. S(x) /\ ~P(x))>>;
   <<(forall x. M(x) ==> ~P(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
     (forall x. S(x) ==> P(x))>>;
   <<(forall x. M(x) ==> ~P(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
     (forall x. S(x) ==> ~P(x))>>;
   <<(forall x. M(x) ==> ~P(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
     (exists x. S(x) /\ P(x))>>;
   <<(forall x. M(x) ==> ~P(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
     (exists x. S(x) /\ ~P(x))>>;
   <<(forall x. P(x) ==> ~M(x)) /\ (forall x. S(x) ==> M(x)) ==>
     (forall x. S(x) ==> P(x))>>;
   <<(forall x. P(x) ==> ~M(x)) /\ (forall x. S(x) ==> M(x)) ==>
     (forall x. S(x) ==> ~P(x))>>;
   <<(forall x. P(x) ==> ~M(x)) /\ (forall x. S(x) ==> M(x)) ==>
     (exists x. S(x) /\ P(x))>>;
   <<(forall x. P(x) ==> ~M(x)) /\ (forall x. S(x) ==> M(x)) ==>
     (exists x. S(x) /\ ~P(x))>>;
   <<(forall x. P(x) ==> ~M(x)) /\ (forall x. M(x) ==> S(x)) ==>
     (forall x. S(x) ==> P(x))>>;
   <<(forall x. P(x) ==> ~M(x)) /\ (forall x. M(x) ==> S(x)) ==>
     (forall x. S(x) ==> ~P(x))>>;
   <<(forall x. P(x) ==> ~M(x)) /\ (forall x. M(x) ==> S(x)) ==>
     (exists x. S(x) /\ P(x))>>;
   <<(forall x. P(x) ==> ~M(x)) /\ (forall x. M(x) ==> S(x)) ==>
     (exists x. S(x) /\ ~P(x))>>;
   <<(forall x. P(x) ==> ~M(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
     (forall x. S(x) ==> P(x))>>;
   <<(forall x. P(x) ==> ~M(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
     (forall x. S(x) ==> ~P(x))>>;
   <<(forall x. P(x) ==> ~M(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
     (exists x. S(x) /\ P(x))>>;
   <<(forall x. P(x) ==> ~M(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
     (exists x. S(x) /\ ~P(x))>>;
   <<(forall x. P(x) ==> ~M(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
     (forall x. S(x) ==> P(x))>>;
   <<(forall x. P(x) ==> ~M(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
     (forall x. S(x) ==> ~P(x))>>;
   <<(forall x. P(x) ==> ~M(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
     (exists x. S(x) /\ P(x))>>;
   <<(forall x. P(x) ==> ~M(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
     (exists x. S(x) /\ ~P(x))>>;
   <<(forall x. P(x) ==> ~M(x)) /\ (exists x. S(x) /\ M(x)) ==>
     (forall x. S(x) ==> P(x))>>;
   <<(forall x. P(x) ==> ~M(x)) /\ (exists x. S(x) /\ M(x)) ==>
     (forall x. S(x) ==> ~P(x))>>;
   <<(forall x. P(x) ==> ~M(x)) /\ (exists x. S(x) /\ M(x)) ==>
     (exists x. S(x) /\ P(x))>>;
   <<(forall x. P(x) ==> ~M(x)) /\ (exists x. S(x) /\ M(x)) ==>
     (exists x. S(x) /\ ~P(x))>>;
   <<(forall x. P(x) ==> ~M(x)) /\ (exists x. M(x) /\ S(x)) ==>
     (forall x. S(x) ==> P(x))>>;
   <<(forall x. P(x) ==> ~M(x)) /\ (exists x. M(x) /\ S(x)) ==>
     (forall x. S(x) ==> ~P(x))>>;
```

```
  <<(forall x. P(x) ==> ~M(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(forall x. P(x) ==> ~M(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(forall x. P(x) ==> ~M(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(forall x. P(x) ==> ~M(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(forall x. P(x) ==> ~M(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(forall x. P(x) ==> ~M(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(forall x. P(x) ==> ~M(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(forall x. P(x) ==> ~M(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(forall x. P(x) ==> ~M(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(forall x. P(x) ==> ~M(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. M(x) /\ P(x)) /\ (forall x. S(x) ==> M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. M(x) /\ P(x)) /\ (forall x. S(x) ==> M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. M(x) /\ P(x)) /\ (forall x. S(x) ==> M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. M(x) /\ P(x)) /\ (forall x. S(x) ==> M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. M(x) /\ P(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. M(x) /\ P(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. M(x) /\ P(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. M(x) /\ P(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. M(x) /\ P(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. M(x) /\ P(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. M(x) /\ P(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. M(x) /\ P(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. M(x) /\ P(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. M(x) /\ P(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. M(x) /\ P(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. M(x) /\ P(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. M(x) /\ P(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. M(x) /\ P(x)) /\ (exists x. S(x) /\ M(x)) ==>
```

```
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. M(x) /\ P(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. M(x) /\ P(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. M(x) /\ P(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. M(x) /\ P(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. M(x) /\ P(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. M(x) /\ P(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. M(x) /\ P(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. M(x) /\ P(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. M(x) /\ P(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. M(x) /\ P(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. M(x) /\ P(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. M(x) /\ P(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. M(x) /\ P(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. M(x) /\ P(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x) /\ M(x)) /\ (forall x. S(x) ==> M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x) /\ M(x)) /\ (forall x. S(x) ==> M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x) /\ M(x)) /\ (forall x. S(x) ==> M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x) /\ M(x)) /\ (forall x. S(x) ==> M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x) /\ M(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x) /\ M(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x) /\ M(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x) /\ M(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x) /\ M(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x) /\ M(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x) /\ M(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x) /\ M(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x) /\ M(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x) /\ M(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x) /\ M(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x) /\ M(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x) /\ M(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x) /\ M(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x) /\ M(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x) /\ M(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x) /\ M(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x) /\ M(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x) /\ M(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x) /\ M(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x) /\ M(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x) /\ M(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x) /\ M(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x) /\ M(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x) /\ M(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x) /\ M(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x) /\ M(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x) /\ M(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. M(x) /\ ~P(x)) /\ (forall x. S(x) ==> M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. M(x) /\ ~P(x)) /\ (forall x. S(x) ==> M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. M(x) /\ ~P(x)) /\ (forall x. S(x) ==> M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. M(x) /\ ~P(x)) /\ (forall x. S(x) ==> M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. M(x) /\ ~P(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. M(x) /\ ~P(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. M(x) /\ ~P(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. M(x) /\ ~P(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. M(x) /\ ~P(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. M(x) /\ ~P(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. M(x) /\ ~P(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. M(x) /\ ~P(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. M(x) /\ ~P(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. M(x) /\ ~P(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. M(x) /\ ~P(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. M(x) /\ ~P(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. M(x) /\ ~P(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. M(x) /\ ~P(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. M(x) /\ ~P(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. M(x) /\ ~P(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. M(x) /\ ~P(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. M(x) /\ ~P(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. M(x) /\ ~P(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. M(x) /\ ~P(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. M(x) /\ ~P(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. M(x) /\ ~P(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. M(x) /\ ~P(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. M(x) /\ ~P(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. M(x) /\ ~P(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. M(x) /\ ~P(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. M(x) /\ ~P(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. M(x) /\ ~P(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x) /\ ~M(x)) /\ (forall x. S(x) ==> M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x) /\ ~M(x)) /\ (forall x. S(x) ==> M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x) /\ ~M(x)) /\ (forall x. S(x) ==> M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x) /\ ~M(x)) /\ (forall x. S(x) ==> M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x) /\ ~M(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x) /\ ~M(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x) /\ ~M(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x) /\ ~M(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x) /\ ~M(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x) /\ ~M(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x) /\ ~M(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x) /\ ~M(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x) /\ ~M(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x) /\ ~M(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x) /\ ~M(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x) /\ ~M(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x) /\ ~M(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x) /\ ~M(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x) /\ ~M(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x) /\ ~M(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x) /\ ~M(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x) /\ ~M(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x) /\ ~M(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x) /\ ~M(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x) /\ ~M(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x) /\ ~M(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x) /\ ~M(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x) /\ ~M(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x) /\ ~M(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x) /\ ~M(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x) /\ ~M(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x) /\ ~M(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
```

```
     (exists x. S(x) /\ ~P(x))>>]
- : dummy_interactive = START_INTERACTIVE
// decidable.p018
val all_valid_syllogisms : fol formula list =
  [<<(forall x. M(x) ==> P(x)) /\ (forall x. S(x) ==> M(x)) ==>
     (forall x. S(x) ==> P(x))>>;
   <<(forall x. M(x) ==> P(x)) /\ (exists x. S(x) /\ M(x)) ==>
     (exists x. S(x) /\ P(x))>>;
   <<(forall x. M(x) ==> P(x)) /\ (exists x. M(x) /\ S(x)) ==>
     (exists x. S(x) /\ P(x))>>;
   <<(forall x. P(x) ==> M(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
     (forall x. S(x) ==> ~P(x))>>;
   <<(forall x. P(x) ==> M(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
     (forall x. S(x) ==> ~P(x))>>;
   <<(forall x. P(x) ==> M(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
     (exists x. S(x) /\ ~P(x))>>;
   <<(forall x. M(x) ==> ~P(x)) /\ (forall x. S(x) ==> M(x)) ==>
     (forall x. S(x) ==> ~P(x))>>;
   <<(forall x. M(x) ==> ~P(x)) /\ (exists x. S(x) /\ M(x)) ==>
     (exists x. S(x) /\ ~P(x))>>;
   <<(forall x. M(x) ==> ~P(x)) /\ (exists x. M(x) /\ S(x)) ==>
     (exists x. S(x) /\ ~P(x))>>;
   <<(forall x. P(x) ==> ~M(x)) /\ (forall x. S(x) ==> M(x)) ==>
     (forall x. S(x) ==> ~P(x))>>;
   <<(forall x. P(x) ==> ~M(x)) /\ (exists x. S(x) /\ M(x)) ==>
     (exists x. S(x) /\ ~P(x))>>;
   <<(forall x. P(x) ==> ~M(x)) /\ (exists x. M(x) /\ S(x)) ==>
     (exists x. S(x) /\ ~P(x))>>;
   <<(exists x. M(x) /\ P(x)) /\ (forall x. M(x) ==> S(x)) ==>
     (exists x. S(x) /\ P(x))>>;
   <<(exists x. P(x) /\ M(x)) /\ (forall x. M(x) ==> S(x)) ==>
     (exists x. S(x) /\ P(x))>>;
   <<(exists x. M(x) /\ ~P(x)) /\ (forall x. M(x) ==> S(x)) ==>
     (exists x. S(x) /\ ~P(x))>>]
// decidable.p019
- : int = 15
// decidable.p020
- : string list =
["If all M are P and all S are M, then all S are P";
 "If all M are P and some S are M, then some S are P";
 "If all M are P and some M are S, then some S are P";
 "If all P are M and no S are M, then no S are P";
 "If all P are M and no M are S, then no S are P";
 "If all P are M and some S are not M, then some S are not P";
 "If no M are P and all S are M, then no S are P";
 "If no M are P and some S are M, then some S are not P";
 "If no M are P and some M are S, then some S are not P";
 "If no P are M and all S are M, then no S are P";
 "If no P are M and some S are M, then some S are not P";
 "If no P are M and some M are S, then some S are not P";
 "If some M are P and all M are S, then some S are P";
 "If some P are M and all M are S, then some S are P";
 "If some M are not P and all M are S, then some S are not P"]
- : dummy_interactive = END_INTERACTIVE
val all_possible_syllogisms' : fol formula list =
  [<<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (forall x. M(x) ==> P(x)) /\ (forall x. S(x) ==> M(x)) ==>
     (forall x. S(x) ==> P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (forall x. M(x) ==> P(x)) /\ (forall x. S(x) ==> M(x)) ==>
     (forall x. S(x) ==> ~P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (forall x. M(x) ==> P(x)) /\ (forall x. S(x) ==> M(x)) ==>
     (exists x. S(x) /\ P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (forall x. M(x) ==> P(x)) /\ (forall x. S(x) ==> M(x)) ==>
     (exists x. S(x) /\ ~P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (forall x. M(x) ==> P(x)) /\ (forall x. M(x) ==> S(x)) ==>
     (forall x. S(x) ==> P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (forall x. M(x) ==> P(x)) /\ (forall x. M(x) ==> S(x)) ==>
     (forall x. S(x) ==> ~P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (forall x. M(x) ==> P(x)) /\ (forall x. M(x) ==> S(x)) ==>
     (exists x. S(x) /\ P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (forall x. M(x) ==> P(x)) /\ (forall x. M(x) ==> S(x)) ==>
     (exists x. S(x) /\ ~P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (forall x. M(x) ==> P(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
     (forall x. S(x) ==> P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (forall x. M(x) ==> P(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
     (forall x. S(x) ==> ~P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (forall x. M(x) ==> P(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
     (exists x. S(x) /\ P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (forall x. M(x) ==> P(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
     (exists x. S(x) /\ ~P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (forall x. M(x) ==> P(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
     (forall x. S(x) ==> P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (forall x. M(x) ==> P(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
     (forall x. S(x) ==> ~P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (forall x. M(x) ==> P(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
     (exists x. S(x) /\ P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (forall x. M(x) ==> P(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
     (exists x. S(x) /\ ~P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (forall x. M(x) ==> P(x)) /\ (exists x. S(x) /\ M(x)) ==>
     (forall x. S(x) ==> P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (forall x. M(x) ==> P(x)) /\ (exists x. S(x) /\ M(x)) ==>
     (forall x. S(x) ==> ~P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
```

```
    (forall x. M(x) ==> P(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. M(x) ==> P(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. M(x) ==> P(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. M(x) ==> P(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. M(x) ==> P(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. M(x) ==> P(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. M(x) ==> P(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. M(x) ==> P(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. M(x) ==> P(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. M(x) ==> P(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. M(x) ==> P(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. M(x) ==> P(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. M(x) ==> P(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. M(x) ==> P(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> M(x)) /\ (forall x. S(x) ==> M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> M(x)) /\ (forall x. S(x) ==> M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> M(x)) /\ (forall x. S(x) ==> M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> M(x)) /\ (forall x. S(x) ==> M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> M(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> M(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> M(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> M(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> M(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> M(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> M(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> M(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> M(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> M(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> M(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> M(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> M(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> M(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> M(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> M(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> M(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> M(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> M(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> M(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> M(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> M(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> M(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> M(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> M(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> M(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> M(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> M(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. M(x) ==> ~P(x)) /\ (forall x. S(x) ==> M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. M(x) ==> ~P(x)) /\ (forall x. S(x) ==> M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. M(x) ==> ~P(x)) /\ (forall x. S(x) ==> M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. M(x) ==> ~P(x)) /\ (forall x. S(x) ==> M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. M(x) ==> ~P(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. M(x) ==> ~P(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. M(x) ==> ~P(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. M(x) ==> ~P(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. M(x) ==> ~P(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
```

```
    (forall x. M(x) ==> ~P(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. M(x) ==> ~P(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. M(x) ==> ~P(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. M(x) ==> ~P(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. M(x) ==> ~P(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. M(x) ==> ~P(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. M(x) ==> ~P(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. M(x) ==> ~P(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. M(x) ==> ~P(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. M(x) ==> ~P(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. M(x) ==> ~P(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. M(x) ==> ~P(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. M(x) ==> ~P(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. M(x) ==> ~P(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. M(x) ==> ~P(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. M(x) ==> ~P(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. M(x) ==> ~P(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. M(x) ==> ~P(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. M(x) ==> ~P(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
```

```
      (exists x. S(x) /\ ~P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (forall x. M(x) ==> ~P(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
      (forall x. S(x) ==> P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (forall x. M(x) ==> ~P(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
      (forall x. S(x) ==> ~P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (forall x. M(x) ==> ~P(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
      (exists x. S(x) /\ P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (forall x. M(x) ==> ~P(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
      (exists x. S(x) /\ ~P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (forall x. P(x) ==> ~M(x)) /\ (forall x. S(x) ==> M(x)) ==>
      (forall x. S(x) ==> P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (forall x. P(x) ==> ~M(x)) /\ (forall x. S(x) ==> M(x)) ==>
      (forall x. S(x) ==> ~P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (forall x. P(x) ==> ~M(x)) /\ (forall x. S(x) ==> M(x)) ==>
      (exists x. S(x) /\ P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (forall x. P(x) ==> ~M(x)) /\ (forall x. S(x) ==> M(x)) ==>
      (exists x. S(x) /\ ~P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (forall x. P(x) ==> ~M(x)) /\ (forall x. M(x) ==> S(x)) ==>
      (forall x. S(x) ==> P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (forall x. P(x) ==> ~M(x)) /\ (forall x. M(x) ==> S(x)) ==>
      (forall x. S(x) ==> ~P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (forall x. P(x) ==> ~M(x)) /\ (forall x. M(x) ==> S(x)) ==>
      (exists x. S(x) /\ P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (forall x. P(x) ==> ~M(x)) /\ (forall x. M(x) ==> S(x)) ==>
      (exists x. S(x) /\ ~P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (forall x. P(x) ==> ~M(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
      (forall x. S(x) ==> P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (forall x. P(x) ==> ~M(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
      (forall x. S(x) ==> ~P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (forall x. P(x) ==> ~M(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
      (exists x. S(x) /\ P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (forall x. P(x) ==> ~M(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
      (exists x. S(x) /\ ~P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (forall x. P(x) ==> ~M(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
      (forall x. S(x) ==> P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (forall x. P(x) ==> ~M(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
      (forall x. S(x) ==> ~P(x))>>;
```

```
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> ~M(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> ~M(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> ~M(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> ~M(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> ~M(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> ~M(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> ~M(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> ~M(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> ~M(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> ~M(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> ~M(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> ~M(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> ~M(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> ~M(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> ~M(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> ~M(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> ~M(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (forall x. P(x) ==> ~M(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
```

```
    (exists x. M(x) /\ P(x)) /\ (forall x. S(x) ==> M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. M(x) /\ P(x)) /\ (forall x. S(x) ==> M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. M(x) /\ P(x)) /\ (forall x. S(x) ==> M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. M(x) /\ P(x)) /\ (forall x. S(x) ==> M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. M(x) /\ P(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. M(x) /\ P(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. M(x) /\ P(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. M(x) /\ P(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. M(x) /\ P(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. M(x) /\ P(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. M(x) /\ P(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. M(x) /\ P(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. M(x) /\ P(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. M(x) /\ P(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. M(x) /\ P(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. M(x) /\ P(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. M(x) /\ P(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. M(x) /\ P(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. M(x) /\ P(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. M(x) /\ P(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. M(x) /\ P(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. M(x) /\ P(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. M(x) /\ P(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. M(x) /\ P(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. M(x) /\ P(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. M(x) /\ P(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. M(x) /\ P(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. M(x) /\ P(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. M(x) /\ P(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. M(x) /\ P(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. M(x) /\ P(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. M(x) /\ P(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. P(x) /\ M(x)) /\ (forall x. S(x) ==> M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. P(x) /\ M(x)) /\ (forall x. S(x) ==> M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. P(x) /\ M(x)) /\ (forall x. S(x) ==> M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. P(x) /\ M(x)) /\ (forall x. S(x) ==> M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. P(x) /\ M(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. P(x) /\ M(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. P(x) /\ M(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. P(x) /\ M(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. P(x) /\ M(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. P(x) /\ M(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. P(x) /\ M(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. P(x) /\ M(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. P(x) /\ M(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. P(x) /\ M(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. P(x) /\ M(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. P(x) /\ M(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. P(x) /\ M(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. P(x) /\ M(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. P(x) /\ M(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. P(x) /\ M(x)) /\ (exists x. S(x) /\ M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. P(x) /\ M(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. P(x) /\ M(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. P(x) /\ M(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
```

```
    (exists x. P(x) /\ M(x)) /\ (exists x. M(x) /\ S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. P(x) /\ M(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. P(x) /\ M(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. P(x) /\ M(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. P(x) /\ M(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. P(x) /\ M(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. P(x) /\ M(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. P(x) /\ M(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. P(x) /\ M(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. M(x) /\ ~P(x)) /\ (forall x. S(x) ==> M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. M(x) /\ ~P(x)) /\ (forall x. S(x) ==> M(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. M(x) /\ ~P(x)) /\ (forall x. S(x) ==> M(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. M(x) /\ ~P(x)) /\ (forall x. S(x) ==> M(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. M(x) /\ ~P(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. M(x) /\ ~P(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (forall x. S(x) ==> ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. M(x) /\ ~P(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (exists x. S(x) /\ P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. M(x) /\ ~P(x)) /\ (forall x. M(x) ==> S(x)) ==>
    (exists x. S(x) /\ ~P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. M(x) /\ ~P(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
    (forall x. S(x) ==> P(x))>>;
  <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
    (exists x. M(x) /\ ~P(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
```

```
      (forall x. S(x) ==> ~P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (exists x. M(x) /\ ~P(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
      (exists x. S(x) /\ P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (exists x. M(x) /\ ~P(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
      (exists x. S(x) /\ ~P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (exists x. M(x) /\ ~P(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
      (forall x. S(x) ==> P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (exists x. M(x) /\ ~P(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
      (forall x. S(x) ==> ~P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (exists x. M(x) /\ ~P(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
      (exists x. S(x) /\ P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (exists x. M(x) /\ ~P(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
      (exists x. S(x) /\ ~P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (exists x. M(x) /\ ~P(x)) /\ (exists x. S(x) /\ M(x)) ==>
      (forall x. S(x) ==> P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (exists x. M(x) /\ ~P(x)) /\ (exists x. S(x) /\ M(x)) ==>
      (forall x. S(x) ==> ~P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (exists x. M(x) /\ ~P(x)) /\ (exists x. S(x) /\ M(x)) ==>
      (exists x. S(x) /\ P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (exists x. M(x) /\ ~P(x)) /\ (exists x. S(x) /\ M(x)) ==>
      (exists x. S(x) /\ ~P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (exists x. M(x) /\ ~P(x)) /\ (exists x. M(x) /\ S(x)) ==>
      (forall x. S(x) ==> P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (exists x. M(x) /\ ~P(x)) /\ (exists x. M(x) /\ S(x)) ==>
      (forall x. S(x) ==> ~P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (exists x. M(x) /\ ~P(x)) /\ (exists x. M(x) /\ S(x)) ==>
      (exists x. S(x) /\ P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (exists x. M(x) /\ ~P(x)) /\ (exists x. M(x) /\ S(x)) ==>
      (exists x. S(x) /\ ~P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (exists x. M(x) /\ ~P(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
      (forall x. S(x) ==> P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (exists x. M(x) /\ ~P(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
      (forall x. S(x) ==> ~P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (exists x. M(x) /\ ~P(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
      (exists x. S(x) /\ P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (exists x. M(x) /\ ~P(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
      (exists x. S(x) /\ ~P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (exists x. M(x) /\ ~P(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
      (forall x. S(x) ==> P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (exists x. M(x) /\ ~P(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
      (forall x. S(x) ==> ~P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (exists x. M(x) /\ ~P(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
      (exists x. S(x) /\ P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (exists x. M(x) /\ ~P(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
      (exists x. S(x) /\ ~P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (exists x. P(x) /\ ~M(x)) /\ (forall x. S(x) ==> M(x)) ==>
      (forall x. S(x) ==> P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (exists x. P(x) /\ ~M(x)) /\ (forall x. S(x) ==> M(x)) ==>
      (forall x. S(x) ==> ~P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (exists x. P(x) /\ ~M(x)) /\ (forall x. S(x) ==> M(x)) ==>
      (exists x. S(x) /\ P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (exists x. P(x) /\ ~M(x)) /\ (forall x. S(x) ==> M(x)) ==>
      (exists x. S(x) /\ ~P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (exists x. P(x) /\ ~M(x)) /\ (forall x. M(x) ==> S(x)) ==>
      (forall x. S(x) ==> P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (exists x. P(x) /\ ~M(x)) /\ (forall x. M(x) ==> S(x)) ==>
      (forall x. S(x) ==> ~P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (exists x. P(x) /\ ~M(x)) /\ (forall x. M(x) ==> S(x)) ==>
      (exists x. S(x) /\ P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (exists x. P(x) /\ ~M(x)) /\ (forall x. M(x) ==> S(x)) ==>
      (exists x. S(x) /\ ~P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (exists x. P(x) /\ ~M(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
      (forall x. S(x) ==> P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (exists x. P(x) /\ ~M(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
      (forall x. S(x) ==> ~P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (exists x. P(x) /\ ~M(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
      (exists x. S(x) /\ P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (exists x. P(x) /\ ~M(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
      (exists x. S(x) /\ ~P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (exists x. P(x) /\ ~M(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
      (forall x. S(x) ==> P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
      (exists x. P(x) /\ ~M(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
      (forall x. S(x) ==> ~P(x))>>;
    <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
```

```
     (exists x. P(x) /\ ~M(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
     (exists x. S(x) /\ P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (exists x. P(x) /\ ~M(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
     (exists x. S(x) /\ ~P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (exists x. P(x) /\ ~M(x)) /\ (exists x. S(x) /\ M(x)) ==>
     (forall x. S(x) ==> P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (exists x. P(x) /\ ~M(x)) /\ (exists x. S(x) /\ M(x)) ==>
     (forall x. S(x) ==> ~P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (exists x. P(x) /\ ~M(x)) /\ (exists x. S(x) /\ M(x)) ==>
     (exists x. S(x) /\ P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (exists x. P(x) /\ ~M(x)) /\ (exists x. S(x) /\ M(x)) ==>
     (exists x. S(x) /\ ~P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (exists x. P(x) /\ ~M(x)) /\ (exists x. M(x) /\ S(x)) ==>
     (forall x. S(x) ==> P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (exists x. P(x) /\ ~M(x)) /\ (exists x. M(x) /\ S(x)) ==>
     (forall x. S(x) ==> ~P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (exists x. P(x) /\ ~M(x)) /\ (exists x. M(x) /\ S(x)) ==>
     (exists x. S(x) /\ P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (exists x. P(x) /\ ~M(x)) /\ (exists x. M(x) /\ S(x)) ==>
     (exists x. S(x) /\ ~P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (exists x. P(x) /\ ~M(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
     (forall x. S(x) ==> P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (exists x. P(x) /\ ~M(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
     (forall x. S(x) ==> ~P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (exists x. P(x) /\ ~M(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
     (exists x. S(x) /\ P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (exists x. P(x) /\ ~M(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
     (exists x. S(x) /\ ~P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (exists x. P(x) /\ ~M(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
     (forall x. S(x) ==> P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (exists x. P(x) /\ ~M(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
     (forall x. S(x) ==> ~P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (exists x. P(x) /\ ~M(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
     (exists x. S(x) /\ P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (exists x. P(x) /\ ~M(x)) /\ (exists x. M(x) /\ ~S(x)) ==>
     (exists x. S(x) /\ ~P(x))>>]
- : dummy_interactive = START_INTERACTIVE
// decidable.p021
```

```
val all_valid_syllogisms' : fol formula list =
  [<<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (forall x. M(x) ==> P(x)) /\ (forall x. S(x) ==> M(x)) ==>
     (forall x. S(x) ==> P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (forall x. M(x) ==> P(x)) /\ (forall x. S(x) ==> M(x)) ==>
     (exists x. S(x) /\ P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (forall x. M(x) ==> P(x)) /\ (forall x. M(x) ==> S(x)) ==>
     (exists x. S(x) /\ P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (forall x. M(x) ==> P(x)) /\ (exists x. S(x) /\ M(x)) ==>
     (exists x. S(x) /\ P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (forall x. M(x) ==> P(x)) /\ (exists x. M(x) /\ S(x)) ==>
     (exists x. S(x) /\ P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (forall x. P(x) ==> M(x)) /\ (forall x. M(x) ==> S(x)) ==>
     (exists x. S(x) /\ P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (forall x. P(x) ==> M(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
     (forall x. S(x) ==> ~P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (forall x. P(x) ==> M(x)) /\ (forall x. S(x) ==> ~M(x)) ==>
     (exists x. S(x) /\ ~P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (forall x. P(x) ==> M(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
     (forall x. S(x) ==> ~P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (forall x. P(x) ==> M(x)) /\ (forall x. M(x) ==> ~S(x)) ==>
     (exists x. S(x) /\ ~P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (forall x. P(x) ==> M(x)) /\ (exists x. S(x) /\ ~M(x)) ==>
     (exists x. S(x) /\ ~P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (forall x. M(x) ==> ~P(x)) /\ (forall x. S(x) ==> M(x)) ==>
     (forall x. S(x) ==> ~P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (forall x. M(x) ==> ~P(x)) /\ (forall x. S(x) ==> M(x)) ==>
     (exists x. S(x) /\ ~P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (forall x. M(x) ==> ~P(x)) /\ (forall x. M(x) ==> S(x)) ==>
     (exists x. S(x) /\ ~P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (forall x. M(x) ==> ~P(x)) /\ (exists x. S(x) /\ M(x)) ==>
     (exists x. S(x) /\ ~P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (forall x. M(x) ==> ~P(x)) /\ (exists x. M(x) /\ S(x)) ==>
     (exists x. S(x) /\ ~P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (forall x. P(x) ==> ~M(x)) /\ (forall x. S(x) ==> M(x)) ==>
     (forall x. S(x) ==> ~P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (forall x. P(x) ==> ~M(x)) /\ (forall x. S(x) ==> M(x)) ==>
     (exists x. S(x) /\ ~P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (forall x. P(x) ==> ~M(x)) /\ (forall x. M(x) ==> S(x)) ==>
     (exists x. S(x) /\ ~P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (forall x. P(x) ==> ~M(x)) /\ (exists x. S(x) /\ M(x)) ==>
     (exists x. S(x) /\ ~P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (forall x. P(x) ==> ~M(x)) /\ (exists x. M(x) /\ S(x)) ==>
     (exists x. S(x) /\ ~P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (exists x. M(x) /\ P(x)) /\ (forall x. M(x) ==> S(x)) ==>
     (exists x. S(x) /\ P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (exists x. P(x) /\ M(x)) /\ (forall x. M(x) ==> S(x)) ==>
     (exists x. S(x) /\ P(x))>>;
   <<(exists x. P(x)) /\ (exists x. M(x)) /\ (exists x. S(x)) ==>
     (exists x. M(x) /\ ~P(x)) /\ (forall x. M(x) ==> S(x)) ==>
     (exists x. S(x) /\ ~P(x))>>]
// decidable.p022
- : int = 24
// decidable.p023
- : string list =
["If all M are P and all S are M, then all S are P";
 "If all M are P and all S are M, then some S are P";
 "If all M are P and all M are S, then some S are P";
 "If all M are P and some S are M, then some S are P";
 "If all M are P and some M are S, then some S are P";
 "If all P are M and all M are S, then some S are P";
 "If all P are M and no S are M, then no S are P";
 "If all P are M and no S are M, then some S are not P";
 "If all P are M and no M are S, then no S are P";
 "If all P are M and no M are S, then some S are not P";
 "If all P are M and some S are not M, then some S are not P";
 "If no M are P and all S are M, then no S are P";
 "If no M are P and all S are M, then some S are not P";
 "If no M are P and all M are S, then some S are not P";
 "If no M are P and some S are M, then some S are not P";
 "If no M are P and some M are S, then some S are not P";
 "If no P are M and all S are M, then no S are P";
 "If no P are M and all S are M, then some S are not P";
 "If no P are M and all M are S, then some S are not P";
 "If no P are M and some S are M, then some S are not P";
 "If no P are M and some M are S, then some S are not P";
 "If some M are P and all M are S, then some S are P";
 "If some P are M and all M are S, then some S are P";
 "If some M are not P and all M are S, then some S are not P"]
- : dummy_interactive = END_INTERACTIVE
val alltuples : int -> 'a list -> 'a list list = <fun>
val allmappings : 'a list -> 'b list -> ('a -> 'b) list = <fun>
val alldepmappings :
  ('a * 'b) list -> ('b -> 'c list) -> ('a -> 'c) list = <fun>
val allfunctions : 'a list -> int -> ('a list -> 'a) list = <fun>
val allpredicates : 'a list -> int -> ('a list -> bool) list = <fun>
val decide_finite : int -> fol formula -> bool = <fun>
val limmeson : int -> fol formula -> (string, term) func * int * int =
  <fun>
val limited_meson :
  int -> fol formula -> ((string, term) func * int * int) list = <fun>
File "/home/student/Harrison/OCaml/decidable.ml", line 273, characters 8-26:
Warning S: this expression should have type unit.
val decide_fmp : fol formula -> bool = <fun>
- : dummy_interactive = START_INTERACTIVE
// decidable.p024
- : bool = true
// decidable.p025
- : bool = false
- : dummy_interactive = END_INTERACTIVE
val decide_monadic : fol formula -> bool = <fun>
- : dummy_interactive = START_INTERACTIVE
// decidable.p027
- : bool = true
- : dummy_interactive = END_INTERACTIVE
- : dummy_interactive = START_INTERACTIVE
// decidable.p029
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
Searching with depth limit 8
Searching with depth limit 9
Searching with depth limit 0
Searching with depth limit 1
- : int list = [1; 3; 9; 1]
// decidable.p030
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
- : int list = [1; 6; 4]
// decidable.p031
```

```
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
- : int list = [1; 5]
- : dummy_interactive = END_INTERACTIVE
#
```

qelim.ml

```
#use qelim ml";;
```

```
# #use "qelim.ml";;
val qelim :
  (fol formula -> fol formula) -> string -> fol formula -> fol formula =
  <fun>
val lift_qelim :
  (string list -> fol formula -> fol formula) ->
  (fol formula -> fol formula) ->
  (string list -> fol formula -> fol formula) ->
  fol formula -> fol formula = <fun>
val cnnf : (fol formula -> fol formula) -> fol formula -> fol formula =
  <fun>
val lfn_dlo : fol formula -> fol formula = <fun>
File "/home/student/Harrison/OCaml/qelim.ml", line 82, characters 23-60:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
(Exists (_, _)|Forall (_, _)|Iff (_, _)|Imp (_, _)|Or (_, _)|And (_, _)|
Not _|True|False)
File "/home/student/Harrison/OCaml/qelim.ml", line 83, characters 24-53:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
(Exists (_, _)|Forall (_, _)|Iff (_, _)|Imp (_, _)|Or (_, _)|And (_, _)|
Not _|True|False)
File "/home/student/Harrison/OCaml/qelim.ml", line 84, characters 24-53:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
(Exists (_, _)|Forall (_, _)|Iff (_, _)|Imp (_, _)|Or (_, _)|And (_, _)|
Not _|True|False)
File "/home/student/Harrison/OCaml/qelim.ml", line 72, characters 2-701:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
(Forall (_, _)|Iff (_, _)|Imp (_, _)|Or (_, _)|And (_, _)|Not _|Atom _|True|
False)
val dlobasic : fol formula -> fol formula = <fun>
val afn_dlo : 'a -> fol formula -> fol formula = <fun>
val quelim_dlo : fol formula -> fol formula = <fun>
- : dummy_interactive = START_INTERACTIVE
// qelim.p001
- : fol formula = <<true>>
// qelim.p002
- : fol formula = <<true>>
// qelim.p003
- : fol formula = <<x < y>>
// qelim.p004
- : fol formula = <<~(b < a \/ b < a)>>
// qelim.p005
- : fol formula = <<true>>
// qelim.p006
```

```
- : fol formula = <<true>>
// qelim.p007
- : fol formula = <<false>>
// qelim.p008
CPU time (user): 0.
- : fol formula = <<true>>
// qelim.p009
CPU time (user): 0.
- : fol formula = <<true>>
// qelim.p010
CPU time (user): 0.
- : fol formula = <<true>>
// qelim.p011
CPU time (user): 0.
- : fol formula = <<false>>
// qelim.p012
CPU time (user): 0.
- : fol formula = <<false>>
// qelim.p013
CPU time (user): 0.
- : fol formula = <<x < y>>
// qelim.p014
CPU time (user): 0.
- : fol formula = <<true>>
// qelim.p015
CPU time (user): 0.
- : fol formula = <<true>>
// qelim.p016
CPU time (user): 0.
- : fol formula = <<true>>
// qelim.p017
CPU time (user): 0.
- : fol formula = <<true>>
// qelim.p018
CPU time (user): 0.
- : fol formula = <<true>>
// qelim.p019
CPU time (user): 0.
- : fol formula = <<x < y>>
// qelim.p020
CPU time (user): 0.
- : fol formula = <<x < y \/ x < y \/ x < y \/ y = x>>
// qelim.p021
CPU time (user): 0.
- : fol formula = <<x < y \/ x < y>>
// qelim.p022
CPU time (user): 0.
- : fol formula = <<true>>
// qelim.p023
CPU time (user): 0.
- : fol formula = <<~(x < z /\ ~w < z)>>
// qelim.p024
CPU time (user): 0.
- : fol formula = <<false>>
// qelim.p025
```

```
CPU time (user): 0.
- : fol formula = <<x < y>>
// qelim.p026
CPU time (user): 0.
- : fol formula = <<true>>
// qelim.p027
CPU time (user): 0.
- : fol formula = <<~(b < a \/ b < a)>>
// qelim.p028
CPU time (user): 0.
- : fol formula = <<~b < a>>
// qelim.p029
CPU time (user): 0.
- : fol formula = <<true>>
// qelim.p030
CPU time (user): 0.
- : fol formula = <<true>>
// qelim.p031
CPU time (user): 0.
- : fol formula = <<false>>
- : dummy_interactive = END_INTERACTIVE
#
```

cooper.ml

```
#use " cooper.ml";;
```

```
# #use "cooper.ml";;
val zero : term = <<|0|>>
val mk_numeral : num -> term = <fun>
val dest_numeral : term -> num = <fun>
val is_numeral : term -> bool = <fun>
val numeral1 : (num -> num) -> term -> term = <fun>
val numeral2 : (num -> num -> num) -> term -> term -> term = <fun>
val linear_cmul : num -> term -> term = <fun>
val linear_add : string list -> term -> term -> term = <fun>
val linear_neg : term -> term = <fun>
val linear_sub : string list -> term -> term -> term = <fun>
val linear_mul : term -> term -> term = <fun>
val lint : string list -> term -> term = <fun>
val mkatom : string list -> string -> term -> fol formula = <fun>
val linform : string list -> fol formula -> fol formula = <fun>
val posineq : fol formula -> fol formula = <fun>
val formlcm : term -> fol formula -> num = <fun>
val adjustcoeff : term -> num -> fol formula -> fol formula = <fun>
val unitycoeff : term -> fol formula -> fol formula = <fun>
val minusinf : term -> fol formula -> fol formula = <fun>
val divlcm : term -> fol formula -> num = <fun>
val bset : term -> fol formula -> term list = <fun>
val linrep : string list -> term -> term -> fol formula -> fol formula =
  <fun>
val cooper : string list -> fol formula -> fol formula = <fun>
val operations : (string * (num -> num -> bool)) list =
  [("=", <fun>); ("<", <fun>); (">", <fun>); ("<=", <fun>);
   (">=", <fun>); ("divides", <fun>)]
File "/home/student/Harrison/OCaml/cooper.ml", line 236, characters 3-167:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
R (_, [])
val evalc : fol formula -> fol formula = <fun>
val integer_qelim : fol formula -> fol formula = <fun>
- : dummy_interactive = START_INTERACTIVE
// cooper.p001
- : fol formula = <<true>>
// cooper.p002
- : fol formula = <<true>>
// cooper.p003
- : fol formula = <<false>>
// cooper.p004
- : fol formula = <<true>>
// cooper.p005
- : fol formula = <<~0 < 1 * a + -1 * b + -1>>
- : dummy_interactive = END_INTERACTIVE
val relativize : (string -> 'a formula) -> 'a formula -> 'a formula =
  <fun>
val natural_qelim : fol formula -> fol formula = <fun>
- : dummy_interactive = START_INTERACTIVE
// cooper.p006
- : fol formula = <<false>>
// cooper.p007
- : fol formula = <<true>>
// cooper.p008
- : fol formula = <<true>>
// cooper.p009
- : fol formula = <<true>>
// Other tests, not in the main text.
// cooper.p010
- : fol formula = <<true>>
// cooper.p011
- : fol formula = <<false>>
// cooper.p012
- : fol formula =
<<~(divides(3,1 * a + 1) /\ 0 < -1 * a + 1 * b + 0 \/
    divides(3,1 * a + 2) /\ 0 < -1 * a + 1 * b + -1 \/
    divides(3,1 * a + 3) /\ 0 < -1 * a + 1 * b + -2)>>
// cooper.p013
CPU time (user): 0.
- : fol formula = <<false>>
// cooper.p014
CPU time (user): 0.00799999999992
- : fol formula = <<divides(65,1 * y + 0) ==> divides(5,1 * y + 0)>>
// cooper.p015
CPU time (user): 0.0120010000001
- : fol formula = <<true>>
// cooper.p016
CPU time (user): 0.
- : fol formula = <<true>>
// cooper.p017
CPU time (user): 0.
- : fol formula = <<true>>
// cooper.p018
CPU time (user): 0.
- : fol formula = <<~0 < -1 * a + 1 * b + 0>>
// cooper.p019
CPU time (user): 0.
- : fol formula = <<~0 < -1 * a + 1 * b + 1>>
// cooper.p020
CPU time (user): 0.00400000000002
- : fol formula = <<false>>
// cooper.p021
CPU time (user): 0.00399999999991
- : fol formula =
<<divides(20,1 * a + 1) /\ 0 < -1 * a + 1 * b + -1 \/
  divides(20,1 * a + 2) /\ 0 < -1 * a + 1 * b + -2 \/
  divides(20,1 * a + 3) /\ 0 < -1 * a + 1 * b + -3 \/
  divides(20,1 * a + 4) /\ 0 < -1 * a + 1 * b + -4 \/
  divides(20,1 * a + 5) /\ 0 < -1 * a + 1 * b + -5 \/
  divides(20,1 * a + 6) /\ 0 < -1 * a + 1 * b + -6 \/
  divides(20,1 * a + 7) /\ 0 < -1 * a + 1 * b + -7 \/
  divides(20,1 * a + 8) /\ 0 < -1 * a + 1 * b + -8 \/
  divides(20,1 * a + 9) /\ 0 < -1 * a + 1 * b + -9 \/
  divides(20,1 * a + 10) /\ 0 < -1 * a + 1 * b + -10 \/
  divides(20,1 * a + 11) /\ 0 < -1 * a + 1 * b + -11 \/
  divides(20,1 * a + 12) /\ 0 < -1 * a + 1 * b + -12 \/
  divides(20,1 * a + 13) /\ 0 < -1 * a + 1 * b + -13 \/
  divides(20,1 * a + 14) /\ 0 < -1 * a + 1 * b + -14 \/
  divides(20,1 * a + 15) /\ 0 < -1 * a + 1 * b + -15 \/
  divides(20,1 * a + 16) /\ 0 < -1 * a + 1 * b + -16 \/
  divides(20,1 * a + 17) /\ 0 < -1 * a + 1 * b + -17 \/
  divides(20,1 * a + 18) /\ 0 < -1 * a + 1 * b + -18 \/
  divides(20,1 * a + 19) /\ 0 < -1 * a + 1 * b + -19 \/
  divides(20,1 * a + 20) /\ 0 < -1 * a + 1 * b + -20>>
// cooper.p022
CPU time (user): 0.00800100000004
- : fol formula = <<false>>
// cooper.p023
CPU time (user): 0.
- : fol formula = <<true>>
// cooper.p024
CPU time (user): 0.00400000000002
- : fol formula = <<false>>
// cooper.p025
CPU time (user): 0.048003
- : fol formula = <<false>>
// cooper.p026
CPU time (user): 0.
- : fol formula = <<true>>
// cooper.p027
CPU time (user): 0.
- : fol formula = <<false>>
// cooper.p028
CPU time (user): 0.016001
- : fol formula = <<true>>
// cooper.p029
CPU time (user): 0.028002
- : fol formula = <<true>>
// cooper.p030
CPU time (user): 0.0120009999999
- : fol formula = <<true>>
// cooper.p031
CPU time (user): 0.00800100000004
- : fol formula = <<true>>
// cooper.p032
CPU time (user): 0.00399999999991
- : fol formula = <<false>>
// cooper.p033
CPU time (user): 0.00800100000004
- : fol formula = <<false>>
// cooper.p034
CPU time (user): 0.0120000000001
- : fol formula = <<true>>
// cooper.p035
CPU time (user): 0.
- : fol formula = <<true>>
```

```
// cooper.p036
CPU time (user): 0.
- : fol formula = <<false>>
// cooper.p037
CPU time (user): 0.00400000000002
- : fol formula = <<true>>
// cooper.p038
CPU time (user): 0.
- : fol formula = <<0 = -6 * x + 5 * y + 0 ==> divides(3,1 * y + 0)>>
// cooper.p039
CPU time (user): 0.076005
- : fol formula = <<true>>
// cooper.p040
CPU time (user): 0.076005
- : fol formula = <<false>>
// cooper.p041
CPU time (user): 0.676042
- : fol formula = <<true>>
// cooper.p042
CPU time (user): 0.00400100000002
- : fol formula = <<true>>
// cooper.p043
CPU time (user): 0.020001
- : fol formula = <<true>>
// cooper.p044
CPU time (user): 0.072005
- : fol formula = <<false>>
// cooper.p045
- : fol formula = <<false>>
// cooper.p046
- : fol formula = <<true>>
// cooper.p047
- : fol formula = <<false>>
// cooper.p048 - Not run
// cooper.p049 - Not run
// cooper.p050
- : fol formula = <<true>>
// cooper.p051
- : fol formula = <<true>>
// cooper.p052
- : fol formula = <<true>>
// cooper.p053 - Not run
// cooper.p054 - Not run
// cooper.p055 - Not run
// cooper.p056
- : fol formula = <<true>>
// cooper.p057 - Not run
- : dummy_interactive = END_INTERACTIVE
#
```

complex.ml

```
#use complex ml";;
```

```
# #use "complex.ml";;
File "/home/student/Harrison/OCaml/complex.ml", line 22, characters 11-106:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
Var _
val poly_add : string list -> term -> term -> term = <fun>
val poly_ladd : string list -> term -> term -> term = <fun>
val poly_neg : term -> term = <fun>
val poly_sub : string list -> term -> term -> term = <fun>
File "/home/student/Harrison/OCaml/complex.ml", line 42, characters 11-199:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
Var _
val poly_mul : string list -> term -> term -> term = <fun>
val poly_lmul : string list -> term -> term -> term = <fun>
val poly_pow : string list -> term -> int -> term = <fun>
val poly_div : string list -> term -> term -> term = <fun>
val poly_var : string -> term = <fun>
val polynate : string list -> term -> term = <fun>
val polyatom : string list -> fol formula -> fol formula = <fun>
- : dummy_interactive = START_INTERACTIVE
// complex.p001
- : fol formula = <<0 = 0>>
- : dummy_interactive = END_INTERACTIVE
val coefficients : string list -> term -> term list = <fun>
val degree : string list -> term -> int = <fun>
val is_constant : string list -> term -> bool = <fun>
val head : string list -> term -> term = <fun>
val behead : string list -> term -> term = <fun>
val poly_cmul : num -> term -> term = <fun>
File "/home/student/Harrison/OCaml/complex.ml", line 123, characters 2-96:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
Var _
val headconst : term -> num = <fun>
val monic : term -> term * bool = <fun>
val pdivide : string list -> term -> term -> int * term = <fun>
type sign = Zero | Nonzero | Positive | Negative
val swap : bool -> sign -> sign = <fun>
val findsign : (term * sign) list -> term -> sign = <fun>
val assertsign :
  (term * sign) list -> term * sign -> (term * sign) list = <fun>
val split_zero :
  (term * sign) list ->
  term ->
  ((term * sign) list -> fol formula) ->
  ((term * sign) list -> fol formula) -> fol formula = <fun>
```

```
val poly_nonzero :
  string list -> (term * sign) list -> term -> fol formula = <fun>
val poly_nondiv :
  string list -> (term * sign) list -> term -> term -> fol formula =
  <fun>
val cqelim :
  string list ->
  term list * term list -> (term * sign) list -> fol formula = <fun>
val init_sgns : (term * sign) list =
  [(<<|1|>>, Positive); (<<|0|>>, Zero)]
File "/home/student/Harrison/OCaml/complex.ml", line 243, characters 29-171:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
(Forall (_, _)|Iff (_, _)|Imp (_, _)|Or (_, _)|And (_, _)|Not _|Atom _|True|
False)
val basic_complex_qelim : string list -> fol formula -> fol formula =
  <fun>
val complex_qelim : fol formula -> fol formula = <fun>
- : dummy_interactive = START_INTERACTIVE
// complex.p002
- : fol formula = <<true>>
// complex.p003
- : fol formula = <<1 + c * (-4 + c * (6 + c * (-4 + c * 1))) = 0>>
// complex.p004
- : fol formula = <<true>>
// complex.p005
- : fol formula = <<true>>
val polytest : term -> term = <fun>
// complex.p006
CPU time (user): 0.0120010000001
val lagrange_4 : term = <<|0|>>
// complex.p007
CPU time (user): 0.064004
val lagrange_8 : term = <<|0|>>
// complex.p008
CPU time (user): 0.016001
val liouville : term = <<|0|>>
// complex.p009
CPU time (user): 0.264017
val fleck : term = <<|0|>>
// complex.p010
CPU time (user): 2.584161
val hurwitz : term = <<|0|>>
// complex.p011
CPU time (user): 5.508344
val schur : term = <<|0|>>
val complex_qelim_all : fol formula -> fol formula = <fun>
// complex.p012
CPU time (user): 0.
- : fol formula = <<true>>
// complex.p013
CPU time (user): 0.
- : fol formula = <<true>>
// complex.p014
```

```
CPU time (user): 0.
- : fol formula = <<true>>
// complex.p015
CPU time (user): 0.
- : fol formula = <<false>>
// complex.p016
CPU time (user): 0.
- : fol formula = <<true>>
// complex.p017
CPU time (user): 0.00800000000004
- : fol formula = <<true>>
// complex.p018
CPU time (user): 0.0120010000001
- : fol formula = <<false>>
// complex.p019
CPU time (user): 0.00800099999992
- : fol formula = <<true>>
// complex.p020
CPU time (user): 0.00400000000002
- : fol formula =
<<(~9 + a * (0 + a * (-10 + a * (0 + a * (5 + a * (0 + a * -1))))) = 0 \/
   ~0 + a *
    (12 + a * (0 + a * (-14 + a * (0 + a * (6 + a * (0 + a * -1)))))) =
    0) /\
  -2 + a * (0 + a * 1) = 0>>
// complex.p021
CPU time (user): 0.00400000000002
- : fol formula =
<<~(~9 + a * (0 + a * (-10 + a * (0 + a * (5 + a * (0 + a * -1))))) = 0 \/
    ~0 + a *
     (12 + a * (0 + a * (-14 + a * (0 + a * (6 + a * (0 + a * -1)))))) =
     0)>>
// complex.p022
CPU time (user): 0.
- : fol formula =
<<~((0 + x * 1 = 0 /\ 1 + x * (0 + x * 1) = 0 \/
     ~0 + x * 1 = 0 /\ 1 + x * (0 + x * (0 + x * (0 + x * 1))) = 0) /\
    ~2 + x * (0 + x * (0 + x * (0 + x * 1))) = 0)>>
// complex.p023
CPU time (user): 0.124008
- : fol formula = <<false>>
// complex.p024 - not run
// complex.p025 - not run
// complex.p026
- : fol formula = <<false>>
// complex.p027
- : fol formula = <<true>>
// complex.p028
CPU time (user): 0.00400100000002
- : fol formula = <<true>>
// complex.p029
CPU time (user): 0.00400000000002
- : fol formula = <<true>>
// complex.p30
CPU time (user): 0.
```

```
- : fol formula = <<true>>
// complex.p31
CPU time (user): 0.0120009999999
- : fol formula = <<true>>
// complex.p032
CPU time (user): 0.0120009999999
- : fol formula = <<true>>
// complex.p033
CPU time (user): 0.
- : fol formula = <<true>>
// complex.p034
CPU time (user): 0.
- : fol formula = <<true>>
// complex.p035
CPU time (user): 0.00400000000002
- : fol formula = <<true>>
// complex.p036
CPU time (user): 0.192012
- : fol formula = <<true>>
// complex.p037
CPU time (user): 0.056003
- : fol formula = <<true>>
// complex.p038
CPU time (user): 0.
- : fol formula = <<true>>
// complex.p039
CPU time (user): 0.020002
- : fol formula = <<true>>
// complex.p40
CPU time (user): 0.00799999999992
- : fol formula =
<<((0 + c * 1) + b * (0 + x * 1)) + a * (0 + x * (0 + x * 1)) = 0 /\
  ((0 + c * 1) + b * (0 + y * 1)) + a * (0 + y * (0 + y * 1)) = 0 /\
  ~(0 + a * 1 = 0 /\
    (0 + b * 1 = 0 /\ 0 + c * 1 = 0 \/
     ~0 + b * 1 = 0 /\
     ~(0 + c * (0 + c * 1)) + b *
      ((0 + c * ((0 + y * 1) + x * 1)) + b * (0 + x * (0 + y * 1))) = 0) \/
    ~0 + a * 1 = 0 /\
    (~(0 + b * (0 + b * (0 + c * -1))) + a *
      (((0 + c * (0 + c * 1)) + b * (0 + c * ((0 + y * -2) + x * -2))) +
       a *
       ((0 + c * ((0 + y * (0 + y * -1)) + x * ((0 + y * -4) + x * -1))) +
        a * (0 + x * (0 + x * (0 + y * (0 + y * 1)))))) =
      0 \/
     ~(0 + b * (0 + b * (0 + b * -1))) + a *
      ((0 + b * ((0 + c * 2) + b * ((0 + y * -2) + x * -2))) + a *
       (((0 + c * ((0 + y * 2) + x * 2)) + b *
         ((0 + y * (0 + y * -1)) + x * ((0 + y * -4) + x * -1))) +
        a * (0 + x * ((0 + y * (0 + y * -2)) + x * (0 + y * -2))))) =
      0)) ==>
  (0 + c * -1) + a * (0 + x * (0 + y * 1)) = 0 /\ (0 + b * 1) + a *
  ((0 + y * 1) + x * 1) = 0>>
// complex.p041
CPU time (user): 0.556035
```

```
- : fol formula =
<<~((~0 + a * 1 = 0 /\
     (0 + a * (0 + a * ((0 + b * -1) + a * (0 + x * -2))) = 0 /\
      (0 + a *
       ((0 + b * (0 + b * -2)) + a *
        (((0 + c * 2) + b * (0 + x * -4)) + a * (0 + x * (0 + x * -2)))) =
       0 /\
       (0 + b * (0 + b * (0 + b * -1))) + a *
       ((0 + b * ((0 + c * 2) + b * (0 + x * -2))) + a *
        ((0 + c * (0 + x * 2)) + b * (0 + x * (0 + x * -1)))) =
       0 /\
       (0 + a * (0 + a * ((0 + c * -1) + a * (0 + x * (0 + x * 1)))) =
        0 /\
        (0 + a *
         ((0 + b * (0 + c * -2)) + a * (0 + c * (0 + x * -4))) = 0 /\
         (0 + b * (0 + b * (0 + c * -1))) + a *
         (((0 + c * (0 + c * 1)) + b * (0 + c * (0 + x * -2))) + a *
          (0 + c * (0 + x * (0 + x * -1)))) =
         0 /\
         (0 + a * 1 = 0 /\
          (0 + b * 1 = 0 /\ 0 + c * 1 = 0 /\ ~0 + a * (0 + x * 1) = 0 \/
           ~0 + b * 1 = 0 /\
           ~(0 + b * (0 + c * -1)) + a * (0 + c * (0 + x * -1)) = 0) \/
          ~0 + a * 1 = 0 /\
          (~0 + a *
            ((0 + c * (0 + c * 1)) + a *
             (0 + c * (0 + x * (0 + x * -1)))) =
            0 \/
           ~0 + a *
            (0 + a *
             ((0 + c * (0 + x * -2)) + b * (0 + x * (0 + x * -1)))) =
            0)) \/
         ~0 + a *
          ((0 + b * (0 + c * -2)) + a * (0 + c * (0 + x * -4))) = 0 /\
         0 + a *
         ((0 + b * (0 + b * (0 + b * (0 + b * (0 + c * (0 + c * -1)))))) +
          a *
          ((0 + b *
            (0 + b *
             ((0 + c * (0 + c * (0 + c * 4))) + b *
              (0 + c * (0 + c * (0 + x * -4)))))) +
           a *
           (((0 + c * (0 + c * (0 + c * (0 + c * 1)))) + b *
             ((0 + c * (0 + c * (0 + c * (0 + x * 16)))) + b *
              (0 + c * (0 + c * (0 + x * (0 + x * -4)))))) +
            a *
            ((0 + c * (0 + c * (0 + c * (0 + x * (0 + x * 14))))) + a *
             (0 + c *
              (0 + c * (0 + x * (0 + x * (0 + x * (0 + x * 1)))))))))) =
         0 /\
         ~0 + a *
          ((0 + b * ((0 + c * (0 + c * 2)) + b * (0 + c * (0 + x * 1)))) +
           a *
           (((0 + c * (0 + c * (0 + x * 3))) + b *
             (0 + c * (0 + x * (0 + x * 2)))) +
            a * (0 + c * (0 + x * (0 + x * (0 + x * 1)))))) =
          0) \/
        ~0 + a * (0 + a * ((0 + c * -1) + a * (0 + x * (0 + x * 1)))) =
         0 /\
        (0 + a *
         (0 + a *
          ((0 + b * (0 + c * 1)) + a *
           ((0 + c * (0 + x * 4)) + b * (0 + x * (0 + x * 1))))) =
         0 /\
         0 + a *
         ((0 + b * (0 + b * (0 + c * 1))) + a *
          (((0 + c * (0 + c * -2)) + b * (0 + c * (0 + x * 2))) + a *
           (0 + c * (0 + x * (0 + x * 2))))) =
         0 /\
         (~0 + a *
           (0 + a *
            (((0 + c * (0 + c * (0 + c * -1))) + b *
              (0 + b * (0 + c * (0 + x * (0 + x * 1))))) +
             a *
             ((0 + b * (0 + c * (0 + x * (0 + x * (0 + x * 2))))) + a *
              (0 + c * (0 + x * (0 + x * (0 + x * (0 + x * 1)))))))) =
           0 \/
          ~0 + a *
           (0 + a *
            (0 + a *
             (((0 + c * (0 + c * (0 + x * 2))) + b *
               (0 + c * (0 + x * (0 + x * 2)))) +
              a * (0 + c * (0 + x * (0 + x * (0 + x * 2))))))) =
           0) \/
         ~0 + a *
          (0 + a *
           ((0 + b * (0 + c * 1)) + a *
            ((0 + c * (0 + x * 4)) + b * (0 + x * (0 + x * 1))))) =
          0 /\
         0 + a *
         (0 + a *
          (0 + a *
           (0 + a *
            (0 + a *
             ((0 + b *
               (0 + b *
                ((0 + c * (0 + c * (0 + c * (0 + c * 1)))) + b *
                 ((0 + c * (0 + c * (0 + c * (0 + x * 2)))) + b *
                  (0 + c * (0 + c * (0 + x * (0 + x * 1)))))))) +
              a *
              (((0 + c * (0 + c * (0 + c * (0 + c * (0 + c * -4))))) +
                b *
                ((0 + c * (0 + c * (0 + c * (0 + c * (0 + x * -8))))) +
                 b *
                 ((0 + c * (0 + c * (0 + c * (0 + x * (0 + x * -3))))) +
                  b *
                  (0 + b *
                   (0 + c * (0 + x * (0 + x * (0 + x * (0 + x * -1))))))))) +
               a *
               (((0 + c *
                   (0 + c * (0 + c * (0 + c * (0 + x * (0 + x * -4)))))) +
                  b *
                  (0 + b *
                   ((0 + c *
                     (0 + c * (0 + x * (0 + x * (0 + x * (0 + x * 3)))))) +
                    b *
                    (0 + c *
                     (0 + x * (0 + x * (0 + x * (0 + x * (0 + x * -2))))))))) +
                 a *
                 (((0 + c *
                    (0 + c *
                     (0 + c * (0 + x * (0 + x * (0 + x * (0 + x * 4))))))) +
                   b *
                   ((0 + c *
                     (0 + c *
                      (0 + x * (0 + x * (0 + x * (0 + x * (0 + x * 8))))))) +
                    b *
                    (0 + c *
                     (0 + x *
                      (0 + x *
                       (0 + x * (0 + x * (0 + x * (0 + x * -1))))))))) +
                  a *
                  (0 + c *
                   (0 + c *
                    (0 + x *
                     (0 + x * (0 + x * (0 + x * (0 + x * (0 + x * 4)))))))))))))))) =
          0 /\
          ~0 + a *
           (0 + a *
            ((0 + b *
              ((0 + c * (0 + c * -1)) + b * (0 + c * (0 + x * -1)))) +
             a *
             (((0 + c * (0 + c * (0 + x * -2))) + b *
               (0 + c * (0 + x * (0 + x * -3)))) +
              a * (0 + c * (0 + x * (0 + x * (0 + x * -2))))))) =
           0)) \/
        ~0 + a *
         ((0 + b * (0 + b * -2)) + a *
          (((0 + c * 2) + b * (0 + x * -4)) + a * (0 + x * (0 + x * -2)))) =
         0 /\
        0 + a *
        (0 + a *
         ((0 + b *
           (0 + b * (0 + b * (0 + b * (0 + b * (0 + b * (0 + c * -1))))))) +
          a *
          ((0 + b *
            (0 + b *
             (0 + b *
              (0 + b *
               ((0 + c * (0 + c * 4)) + b *
                ((0 + c * (0 + x * -4)) + b * (0 + x * (0 + x * 1)))))))) +
           a *
           ((0 + b *
             (0 + b *
              ((0 + c * (0 + c * (0 + c * -8))) + b *
               ((0 + c * (0 + c * (0 + x * 4))) + b *
                ((0 + c * (0 + x * (0 + x * -14))) + b *
                 (0 + x * (0 + x * (0 + x * 4)))))))) +
            a *
            (((0 + c * (0 + c * (0 + c * (0 + c * 4)))) + b *
              ((0 + c * (0 + c * (0 + c * (0 + x * -8)))) + b *
               ((0 + c * (0 + c * (0 + x * (0 + x * -4)))) + b *
                ((0 + c * (0 + x * (0 + x * (0 + x * -32)))) + b *
                 (0 + x * (0 + x * (0 + x * (0 + x * 6)))))))) +
             a *
             ((0 + b *
               ((0 + c * (0 + c * (0 + x * (0 + x * (0 + x * -4))))) +
                b *
                ((0 + c * (0 + x * (0 + x * (0 + x * (0 + x * -37))))) +
                 b * (0 + x * (0 + x * (0 + x * (0 + x * (0 + x * 4)))))))) +
              a *
              ((0 + b *
                ((0 + c *
                  (0 + x * (0 + x * (0 + x * (0 + x * (0 + x * -20)))))) +
                 b *
                 (0 + x *
                  (0 + x * (0 + x * (0 + x * (0 + x * (0 + x * 1)))))))) +
               a *
               (0 + c *
                (0 + x *
                 (0 + x * (0 + x * (0 + x * (0 + x * (0 + x * -4)))))))))))))) =
        0 /\
        0 + a *
        ((0 + b * (0 + b * (0 + b * (0 + b * (0 + b * (0 + b * -1)))))) +
         a *
         ((0 + b *
           (0 + b * (0 + b * (0 + b * ((0 + c * 6) + b * (0 + x * -4)))))) +
          a *
          ((0 + b *
            (0 + b *
             ((0 + c * (0 + c * -8)) + b *
              ((0 + c * (0 + x * 20)) + b * (0 + x * (0 + x * -6)))))) +
           a *
           (((0 + c * (0 + c * (0 + c * 4))) + b *
             ((0 + c * (0 + c * (0 + x * -12))) + b *
              ((0 + c * (0 + x * (0 + x * 26))) + b *
               (0 + x * (0 + x * (0 + x * -4)))))) +
            a *
            (((0 + c * (0 + c * (0 + x * (0 + x * -4)))) + b *
              ((0 + c * (0 + x * (0 + x * (0 + x * 16)))) + b *
               (0 + x * (0 + x * (0 + x * (0 + x * -1)))))) +
             a * (0 + c * (0 + x * (0 + x * (0 + x * (0 + x * 4)))))))))) =
        0 /\
        ~0 + a *
         ((0 + b * (0 + b * ((0 + c * 2) + b * (0 + x * 1)))) + a *
          (((0 + c * (0 + c * -2)) + b *
            ((0 + c * (0 + x * 2)) + b * (0 + x * (0 + x * 2)))) +
           a * (0 + b * (0 + x * (0 + x * (0 + x * 1)))))) =
         0) \/
       ~0 + a * (0 + a * ((0 + b * -1) + a * (0 + x * -2))) = 0 /\
       (0 + a *
        (0 + a *
         (0 + a *
          (0 + a *
           (((0 + c * (0 + c * 2)) + b *
             ((0 + c * (0 + x * 4)) + b * (0 + x * (0 + x * 2)))) +
            a *
            (((0 + c * (0 + x * (0 + x * 4))) + b *
              (0 + x * (0 + x * (0 + x * 4)))) +
             a * (0 + x * (0 + x * (0 + x * (0 + x * 2))))))))) =
        0 /\
        0 + a *
        (0 + a *
         (0 + a *
          ((0 + b *
            ((0 + c * (0 + c * 1)) + b *
             ((0 + c * (0 + x * 2)) + b * (0 + x * (0 + x * 1))))) +
           a *
           ((0 + b *
             ((0 + c * (0 + x * (0 + x * 2))) + b *
              (0 + x * (0 + x * (0 + x * 2))))) +
            a * (0 + b * (0 + x * (0 + x * (0 + x * (0 + x * 1))))))))) =
        0 /\
        (0 + a *
         (0 + a *
          ((0 + b * (0 + b * 1)) + a *
           (((0 + c * -2) + b * (0 + x * 2)) + a * (0 + x * (0 + x * 2))))) =
         0 /\
         0 + a *
         ((0 + b * (0 + b * (0 + b * 1))) + a *
          ((0 + b * ((0 + c * -3) + b * (0 + x * 2))) + a *
           ((0 + c * (0 + x * -4)) + b * (0 + x * (0 + x * 1))))) =
         0 /\
         (~0 + a *
           (0 + a *
            ((0 + b *
              ((0 + c * (0 + c * -1)) + b *
               (0 + b * (0 + x * (0 + x * 1))))) +
             a *
             (((0 + c * (0 + c * (0 + x * -2))) + b *
               ((0 + c * (0 + x * (0 + x * -2))) + b *
                (0 + x * (0 + x * (0 + x * 2))))) +
              a *
              ((0 + c * (0 + x * (0 + x * (0 + x * -2)))) + b *
               (0 + x * (0 + x * (0 + x * (0 + x * 1)))))))) =
           0 \/
          ~0 + a *
           (0 + a *
            (0 + a *
             ((0 + b *
               ((0 + c * (0 + x * 2)) + b * (0 + x * (0 + x * 2)))) +
              a *
              (((0 + c * (0 + x * (0 + x * 2))) + b *
                (0 + x * (0 + x * (0 + x * 4)))) +
               a * (0 + x * (0 + x * (0 + x * (0 + x * 2)))))))) =
           0) \/
         ~0 + a *
          (0 + a *
           ((0 + b * (0 + b * 1)) + a *
            (((0 + c * -2) + b * (0 + x * 2)) + a *
             (0 + x * (0 + x * 2))))) =
          0 /\
         0 + a *
         (0 + a *
          (0 + a *
           (0 + a *
            (0 + a *
             (0 + a *
              ((0 + b *
                (0 + b *
                 (0 + b *
                  ((0 + c * (0 + c * 1)) + b *
                   ((0 + c * (0 + x * 2)) + b * (0 + x * (0 + x * 1))))))) +
               a *
               ((0 + b *
                 ((0 + c * (0 + c * (0 + c * -4))) + b *
                  ((0 + c * (0 + c * (0 + x * -6))) + b *
                   ((0 + c * (0 + x * (0 + x * 2))) + b *
                    (0 + x * (0 + x * (0 + x * 4))))))) +
                a *
                (((0 + c * (0 + c * (0 + c * (0 + x * -8)))) + b *
                  ((0 + c * (0 + c * (0 + x * (0 + x * -24)))) + b *
                   ((0 + c * (0 + x * (0 + x * (0 + x * -12)))) + b *
                    (0 + x * (0 + x * (0 + x * (0 + x * 5))))))) +
                 a *
                 (((0 + c * (0 + c * (0 + x * (0 + x * (0 + x * -16))))) +
                   b *
                   ((0 + c * (0 + x * (0 + x * (0 + x * (0 + x * -20))))) +
                    b *
                    (0 + x * (0 + x * (0 + x * (0 + x * (0 + x * 2))))))) +
                  a *
                  (0 + c *
                   (0 + x * (0 + x * (0 + x * (0 + x * (0 + x * -8))))))))))))))) =
         0 /\
         ~0 + a *
          (0 + a *
           ((0 + b * (0 + b * ((0 + c * -1) + b * (0 + x * -1)))) + a *
            (((0 + c * (0 + c * 2)) + b *
              ((0 + c * (0 + x * 1)) + b * (0 + x * (0 + x * -2)))) +
             a *
             ((0 + c * (0 + x * (0 + x * 2))) + b *
              (0 + x * (0 + x * (0 + x * -1))))))) =
          0) \/
        ~0 + a *
         (0 + a *
          (0 + a *
           (0 + a *
            (((0 + c * (0 + c * 2)) + b *
              ((0 + c * (0 + x * 4)) + b * (0 + x * (0 + x * 2)))) +
             a *
             (((0 + c * (0 + x * (0 + x * 4))) + b *
               (0 + x * (0 + x * (0 + x * 4)))) +
              a * (0 + x * (0 + x * (0 + x * (0 + x * 2))))))))) =
         0 /\
        0 + a *
        (0 + a *
         (0 + a *
          (0 + a *
           (0 + a *
            (0 + a *
             (0 + a *
              (0 + a *
               ((0 + b *
                 (0 + b *
                  (0 + b *
                   ((0 + c * (0 + c * (0 + c * (0 + c * -1)))) + b *
                    ((0 + c * (0 + c * (0 + c * (0 + x * -4)))) + b *
                     ((0 + c * (0 + c * (0 + x * (0 + x * -6)))) + b *
                      ((0 + c * (0 + x * (0 + x * (0 + x * -4)))) + b *
                       (0 + x * (0 + x * (0 + x * (0 + x * -1))))))))))))) +
                a *
                ((0 + b *
                  ((0 + c * (0 + c * (0 + c * (0 + c * (0 + c * 4))))) +
                   b *
                   ((0 + c * (0 + c * (0 + c * (0 + c * (0 + x * 14))))) +
                    b *
                    ((0 + c * (0 + c * (0 + c * (0 + x * (0 + x * 12))))) +
                     b *
                     ((0 + c *
                       (0 + c * (0 + x * (0 + x * (0 + x * -8))))) +
                      b *
                      ((0 + c *
                        (0 + x * (0 + x * (0 + x * (0 + x * -16))))) +
                       b *
                       (0 + x *
                        (0 + x * (0 + x * (0 + x * (0 + x * -6))))))))))) +
                 a *
                 (((0 + c *
                    (0 + c * (0 + c * (0 + c * (0 + c * (0 + x * 8)))))) +
                   b *
                   ((0 + c *
                     (0 + c * (0 + c * (0 + c * (0 + x * (0 + x * 48)))))) +
                    b *
                    ((0 + c *
                      (0 + c *
                       (0 + c * (0 + x * (0 + x * (0 + x * 88)))))) +
                     b *
                     ((0 + c *
                       (0 + c *
                        (0 + x * (0 + x * (0 + x * (0 + x * 50)))))) +
                      b *
                      ((0 + c *
                        (0 + x *
                         (0 + x * (0 + x * (0 + x * (0 + x * -12)))))) +
                       b *
```

```
                     (0 + x *
                      (0 + x *
                       (0 + x * (0 + x * (0 + x * (0 + x * -14))))))))))) +
                a *
                (((0 + c *
                   (0 + c *
                    (0 + c *
                     (0 + c * (0 + x * (0 + x * (0 + x * 32))))))) +
                  b *
                  ((0 + c *
                    (0 + c *
                     (0 + c *
                      (0 + x * (0 + x * (0 + x * (0 + x * 120))))))) +
                   b *
                   ((0 + c *
                     (0 + c *
                      (0 + x *
                       (0 + x * (0 + x * (0 + x * (0 + x * 132))))))) +
                    b *
                    ((0 + c *
                      (0 + x *
                       (0 + x *
                        (0 + x * (0 + x * (0 + x * (0 + x * 28))))))) +
                     b *
                     (0 + x *
                      (0 + x *
                       (0 + x *
                        (0 + x * (0 + x * (0 + x * (0 + x * -16))))))))))) +
                 a *
                 (((0 + c *
                    (0 + c *
                     (0 + c *
                      (0 + x *
                       (0 + x * (0 + x * (0 + x * (0 + x * 48)))))))) +
                   b *
                   ((0 + c *
                     (0 + c *
                      (0 + x *
                       (0 + x *
                        (0 + x * (0 + x * (0 + x * (0 + x * 112)))))))) +
                    b *
                    ((0 + c *
                      (0 + x *
                       (0 + x *
                        (0 + x *
                         (0 + x * (0 + x * (0 + x * (0 + x * 56)))))))) +
                     b *
                     (0 + x *
                      (0 + x *
                       (0 + x *
                        (0 + x *
                         (0 + x * (0 + x * (0 + x * (0 + x * -9))))))))))) +
                  a *
                  (((0 + c *
                     (0 + c *
                      (0 + x *
                       (0 + x *
                        (0 + x *
                         (0 + x * (0 + x * (0 + x * (0 + x * 32))))))))) +
                    b *
                    ((0 + c *
                      (0 + x *
                       (0 + x *
                        (0 + x *
                         (0 + x *
                          (0 + x * (0 + x * (0 + x * (0 + x * 36))))))))) +
                     b *
                     (0 + x *
                      (0 + x *
                       (0 + x *
                        (0 + x *
                         (0 + x *
                          (0 + x * (0 + x * (0 + x * (0 + x * -2))))))))))) +
                   a *
                   (0 + c *
                    (0 + x *
                     (0 + x *
                      (0 + x *
                       (0 + x *
                        (0 + x *
                         (0 + x * (0 + x * (0 + x * (0 + x * 8))))))))))))))))))))))) =
      0 /\
      0 + a *
      (0 + a *
       (0 + a *
        (0 + a *
         (0 + a *
          ((0 + b *
            (0 + b *
             (0 + b *
              ((0 + c * (0 + c * 1)) + b *
               ((0 + c * (0 + x * 2)) + b * (0 + x * (0 + x * 1))))))) +
           a *
           ((0 + b *
             ((0 + c * (0 + c * (0 + c * -4))) + b *
              ((0 + c * (0 + c * (0 + x * -6))) + b *
               ((0 + c * (0 + x * (0 + x * 2))) + b *
                (0 + x * (0 + x * (0 + x * 4))))))) +
            a *
            (((0 + c * (0 + c * (0 + c * (0 + x * -8)))) + b *
              ((0 + c * (0 + c * (0 + x * (0 + x * -24)))) + b *
               ((0 + c * (0 + x * (0 + x * (0 + x * -12)))) + b *
                (0 + x * (0 + x * (0 + x * (0 + x * 5))))))) +
             a *
             (((0 + c * (0 + c * (0 + x * (0 + x * (0 + x * -16))))) +
               b *
               ((0 + c * (0 + x * (0 + x * (0 + x * (0 + x * -20))))) +
                b *
                (0 + x * (0 + x * (0 + x * (0 + x * (0 + x * 2))))))) +
              a *
```

```
              (0 + c *
               (0 + x * (0 + x * (0 + x * (0 + x * (0 + x * -8))))))))))))))) =
      0 /\
      ~0 + a *
       (0 + a *
        (0 + a *
         (0 + a *
          (((0 + c * (0 + c * (0 + c * -2))) + b *
            ((0 + c * (0 + c * (0 + x * -5))) + b *
             ((0 + c * (0 + x * (0 + x * -4))) + b *
              (0 + x * (0 + x * (0 + x * -1)))))) +
           a *
           (((0 + c * (0 + c * (0 + x * (0 + x * -4)))) + b *
             ((0 + c * (0 + x * (0 + x * (0 + x * -6)))) + b *
              (0 + x * (0 + x * (0 + x * (0 + x * -2)))))) +
            a *
            ((0 + c * (0 + x * (0 + x * (0 + x * (0 + x * -2))))) +
             b * (0 + x * (0 + x * (0 + x * (0 + x * (0 + x * -1))))))))))) =
       0)) \/
    0 + a * 1 = 0 /\
    ~0 + b * 1 = 0 /\
    (0 + b * ((0 + c * 1) + b * (0 + x * 1)) = 0 /\
     (0 + c * (0 + c * 1)) + b * (0 + c * (0 + x * 1)) = 0 /\
     (0 + a * 1 = 0 /\
      (0 + b * 1 = 0 /\ 0 + c * 1 = 0 /\ ~0 + a * (0 + x * 1) = 0 \/
       ~0 + b * 1 = 0 /\
       ~(0 + b * (0 + c * -1)) + a * (0 + c * (0 + x * -1)) = 0) \/
      ~0 + a * 1 = 0 /\
      (~0 + a *
        ((0 + c * (0 + c * 1)) + a * (0 + c * (0 + x * (0 + x * -1)))) =
        0 \/
       ~0 + a *
        (0 + a * ((0 + c * (0 + x * -2)) + b * (0 + x * (0 + x * -1)))) =
        0)) \/
     ~0 + b * ((0 + c * 1) + b * (0 + x * 1)) = 0 /\
     0 + a *
     ((0 + c * (0 + c * (0 + c * (0 + c * 1)))) + b *
      ((0 + c * (0 + c * (0 + c * (0 + x * 2)))) + b *
       (0 + c * (0 + c * (0 + x * (0 + x * 1)))))) =
     0 /\
     ~(0 + b * ((0 + c * (0 + c * -1)) + b * (0 + c * (0 + x * -1)))) +
      a *
      ((0 + c * (0 + c * (0 + x * -1))) + b *
       (0 + c * (0 + x * (0 + x * -1)))) =
      0) \/
    0 + a * 1 = 0 /\
    0 + b * 1 = 0 /\
    ~0 + c * 1 = 0 /\
    (0 + a * 1 = 0 /\
     (0 + b * 1 = 0 /\ 0 + c * 1 = 0 /\ ~0 + a * (0 + x * 1) = 0 \/
      ~0 + b * 1 = 0 /\
      ~(0 + b * (0 + c * -1)) + a * (0 + c * (0 + x * -1)) = 0) \/
     ~0 + a * 1 = 0 /\
     (~0 + a *
       ((0 + c * (0 + c * 1)) + a * (0 + c * (0 + x * (0 + x * -1)))) =
       0 \/
      ~0 + a *
       (0 + a * ((0 + c * (0 + x * -2)) + b * (0 + x * (0 + x * -1)))) =
       0))) /\
   ((0 + c * 1) + b * (0 + x * 1)) + a * (0 + x * (0 + x * 1)) = 0 \/
   (~0 + a * 1 = 0 /\
    (0 + a * (0 + a * ((0 + b * -1) + a * (0 + x * -2))) = 0 /\
     (0 + a *
      ((0 + b * (0 + b * -2)) + a *
       (((0 + c * 2) + b * (0 + x * -4)) + a * (0 + x * (0 + x * -2)))) =
      0 /\
      (0 + b * (0 + b * (0 + b * -1))) + a *
      ((0 + b * ((0 + c * 2) + b * (0 + x * -2))) + a *
       ((0 + c * (0 + x * 2)) + b * (0 + x * (0 + x * -1)))) =
      0 /\
      (0 + a * (0 + a * ((0 + c * -1) + a * (0 + x * (0 + x * 1)))) =
       0 /\
       (0 + a *
        ((0 + b * (0 + c * -2)) + a * (0 + c * (0 + x * -4))) = 0 /\
        (0 + b * (0 + b * (0 + c * -1))) + a *
        (((0 + c * (0 + c * 1)) + b * (0 + c * (0 + x * -2))) + a *
         (0 + c * (0 + x * (0 + x * -1)))) =
        0 /\
        0 + a * 1 = 0 /\
        (0 + b * 1 = 0 /\
         0 + c * 1 = 0 /\ ~(0 + b * 1) + a * (0 + x * 1) = 0 \/
         ~0 + b * 1 = 0 /\
         ~(0 + b * (0 + b * 1)) + a *
          ((0 + c * -1) + b * (0 + x * 1)) = 0) \/
        ~0 + a *
         ((0 + b * (0 + c * -2)) + a * (0 + c * (0 + x * -4))) = 0 /\
        0 + a *
        ((0 + b * (0 + b * (0 + b * (0 + b * (0 + c * (0 + c * -1)))))) +
         a *
         ((0 + b *
           (0 + b *
            ((0 + c * (0 + c * (0 + c * 4))) + b *
             (0 + c * (0 + c * (0 + x * -4)))))) +
          a *
          (((0 + c * (0 + c * (0 + c * (0 + c * 1)))) + b *
            ((0 + c * (0 + c * (0 + c * (0 + x * 16)))) + b *
             (0 + c * (0 + c * (0 + x * (0 + x * -4)))))) +
           a *
           ((0 + c * (0 + c * (0 + c * (0 + x * (0 + x * 14))))) + a *
            (0 + c *
             (0 + c * (0 + x * (0 + x * (0 + x * (0 + x * 1)))))))))) =
        0 /\
        ~0 + a *
         ((0 + b * (0 + b * (0 + c * -1))) + a *
          (((0 + c * (0 + c * -1)) + b * (0 + c * (0 + x * -4))) + a *
           (0 + c * (0 + x * (0 + x * -3))))) =
         0) \/
       ~0 + a * (0 + a * ((0 + c * -1) + a * (0 + x * (0 + x * 1)))) =
        0 /\
       (0 + a *
        (0 + a *
         ((0 + b * (0 + c * 1)) + a *
          ((0 + c * (0 + x * 4)) + b * (0 + x * (0 + x * 1))))) =
        0 /\
        0 + a *
        ((0 + b * (0 + b * (0 + c * 1))) + a *
         (((0 + c * (0 + c * -2)) + b * (0 + c * (0 + x * 2))) + a *
          (0 + c * (0 + x * (0 + x * 2))))) =
        0 /\
        (~0 + a *
          (0 + a *
           (0 + a *
            (((0 + c * (0 + c * -1)) + b *
              (0 + b * (0 + x * (0 + x * 1)))) +
             a *
             ((0 + b * (0 + x * (0 + x * (0 + x * 2)))) + a *
              (0 + x * (0 + x * (0 + x * (0 + x * 1)))))))) =
          0 \/
         ~0 + a *
          (0 + a *
           (0 + a *
            (0 + a *
             (((0 + c * (0 + x * 2)) + b * (0 + x * (0 + x * 2))) +
              a * (0 + x * (0 + x * (0 + x * 2))))))) =
          0) \/
        ~0 + a *
         (0 + a *
          ((0 + b * (0 + c * 1)) + a *
           ((0 + c * (0 + x * 4)) + b * (0 + x * (0 + x * 1))))) =
         0 /\
        0 + a *
        (0 + a *
         (0 + a *
          (0 + a *
           (0 + a *
            ((0 + b *
              (0 + b *
               ((0 + c * (0 + c * (0 + c * (0 + c * 1)))) + b *
                ((0 + c * (0 + c * (0 + c * (0 + x * 2)))) + b *
                 (0 + c * (0 + c * (0 + x * (0 + x * 1)))))))) +
             a *
             (((0 + c * (0 + c * (0 + c * (0 + c * (0 + c * -4))))) +
               b *
               ((0 + c * (0 + c * (0 + c * (0 + c * (0 + x * -8))))) +
                b *
                ((0 + c * (0 + c * (0 + c * (0 + x * (0 + x * -3))))) +
                 b *
                 (0 + b *
                  (0 + c * (0 + x * (0 + x * (0 + x * (0 + x * -1))))))))) +
              a *
              (((0 + c *
                 (0 + c * (0 + c * (0 + c * (0 + x * (0 + x * -4)))))) +
                b *
                (0 + b *
                 ((0 + c *
                     (0 + c * (0 + x * (0 + x * (0 + x * (0 + x * 3)))))) +
                    b *
                    (0 + c *
                     (0 + x * (0 + x * (0 + x * (0 + x * (0 + x * -2))))))))) +
                 a *
                 (((0 + c *
                    (0 + c *
                     (0 + c * (0 + x * (0 + x * (0 + x * (0 + x * 4))))))) +
                   b *
                   ((0 + c *
                     (0 + c *
                      (0 + x * (0 + x * (0 + x * (0 + x * (0 + x * 8))))))) +
                    b *
                    (0 + c *
                     (0 + x *
                      (0 + x *
                       (0 + x * (0 + x * (0 + x * (0 + x * -1))))))))) +
                  a *
                  (0 + c *
                   (0 + c *
                    (0 + x *
                     (0 + x * (0 + x * (0 + x * (0 + x * (0 + x * 4)))))))))))))))) =
          0 /\
          ~0 + a *
           (0 + a *
            (0 + a *
             (((0 + c * (0 + c * 2)) + b *
               ((0 + c * (0 + x * 3)) + b * (0 + x * (0 + x * 1)))) +
              a *
              ((0 + c * (0 + x * (0 + x * 2))) + b *
               (0 + x * (0 + x * (0 + x * 1))))))) =
           0)) \/
        ~0 + a *
         ((0 + b * (0 + b * -2)) + a *
          (((0 + c * 2) + b * (0 + x * -4)) + a * (0 + x * (0 + x * -2)))) =
         0 /\
        0 + a *
        (0 + a *
         ((0 + b *
           (0 + b * (0 + b * (0 + b * (0 + b * (0 + b * (0 + c * -1))))))) +
          a *
          ((0 + b *
            (0 + b *
             (0 + b *
              (0 + b *
               ((0 + c * (0 + c * 4)) + b *
                ((0 + c * (0 + x * -4)) + b * (0 + x * (0 + x * 1)))))))) +
           a *
           ((0 + b *
             (0 + b *
              ((0 + c * (0 + c * (0 + c * -8))) + b *
               ((0 + c * (0 + c * (0 + x * 4))) + b *
                ((0 + c * (0 + x * (0 + x * -14))) + b *
                 (0 + x * (0 + x * (0 + x * 4)))))))) +
            a *
```

```
     (((0 + c * (0 + c * (0 + c * (0 + c * 4)))) + b *
       ((0 + c * (0 + c * (0 + c * (0 + x * -8)))) + b *
        ((0 + c * (0 + c * (0 + x * (0 + x * -4)))) + b *
         ((0 + c * (0 + x * (0 + x * (0 + x * -32)))) + b *
          (0 + x * (0 + x * (0 + x * (0 + x * 6)))))))) +
      a *
      ((0 + b *
        ((0 + c * (0 + c * (0 + x * (0 + x * (0 + x * -4))))) +
         b *
         ((0 + c * (0 + x * (0 + x * (0 + x * (0 + x * -37))))) +
          b * (0 + x * (0 + x * (0 + x * (0 + x * (0 + x * 4)))))))) +
       a *
       ((0 + b *
         ((0 + c *
           (0 + x * (0 + x * (0 + x * (0 + x * (0 + x * -20)))))) +
          b *
          (0 + x *
           (0 + x * (0 + x * (0 + x * (0 + x * (0 + x * 1)))))))) +
        a *
        (0 + c *
         (0 + x *
          (0 + x * (0 + x * (0 + x * (0 + x * (0 + x * -4)))))))))))))) =
 0 /\
 0 + a *
 ((0 + b * (0 + b * (0 + b * (0 + b * (0 + b * (0 + b * -1)))))) +
  a *
  ((0 + b *
    (0 + b * (0 + b * (0 + b * ((0 + c * 6) + b * (0 + x * -4)))))) +
   a *
   ((0 + b *
     (0 + b *
      ((0 + c * (0 + c * -8)) + b *
       ((0 + c * (0 + x * 20)) + b * (0 + x * (0 + x * -6)))))) +
    a *
    (((0 + c * (0 + c * (0 + c * 4))) + b *
      ((0 + c * (0 + c * (0 + x * -12))) + b *
       ((0 + c * (0 + x * (0 + x * 26))) + b *
        (0 + x * (0 + x * (0 + x * -4)))))) +
     a *
     (((0 + c * (0 + c * (0 + x * (0 + x * -4)))) + b *
       ((0 + c * (0 + x * (0 + x * (0 + x * 16)))) + b *
        (0 + x * (0 + x * (0 + x * (0 + x * -1)))))) +
      a * (0 + c * (0 + x * (0 + x * (0 + x * (0 + x * 4)))))))))) =
 0 /\
 ~0 + a *
  ((0 + b * (0 + b * (0 + b * -1))) + a *
   ((0 + b * (0 + b * (0 + x * -4))) + a *
    ((0 + b * (0 + x * (0 + x * -5))) + a *
     (0 + x * (0 + x * (0 + x * -2)))))) =
  0) \/
~0 + a * (0 + a * ((0 + b * -1) + a * (0 + x * -2))) = 0 /\
(0 + a *
 (0 + a *
  (0 + a *
   (0 + a *
    (((0 + c * (0 + c * 2)) + b *
      ((0 + c * (0 + x * 4)) + b * (0 + x * (0 + x * 2)))) +
     a *
     (((0 + c * (0 + x * (0 + x * 4))) + b *
       (0 + x * (0 + x * (0 + x * 4)))) +
      a * (0 + x * (0 + x * (0 + x * (0 + x * 2))))))))) =
 0 /\
 0 + a *
 (0 + a *
  (0 + a *
   ((0 + b *
     ((0 + c * (0 + c * 1)) + b *
      ((0 + c * (0 + x * 2)) + b * (0 + x * (0 + x * 1))))) +
    a *
    ((0 + b *
      ((0 + c * (0 + x * (0 + x * 2))) + b *
       (0 + x * (0 + x * (0 + x * 2))))) +
     a * (0 + b * (0 + x * (0 + x * (0 + x * (0 + x * 1))))))))) =
 0 /\
 (0 + a *
  (0 + a *
   ((0 + b * (0 + b * 1)) + a *
    (((0 + c * -2) + b * (0 + x * 2)) + a * (0 + x * (0 + x * 2))))) =
  0 /\
  0 + a *
  ((0 + b * (0 + b * (0 + b * 1))) + a *
   ((0 + b * ((0 + c * -3) + b * (0 + x * 2))) + a *
    ((0 + c * (0 + x * -4)) + b * (0 + x * (0 + x * 1))))) =
  0 /\
  (~0 + a *
    (0 + a *
     (0 + a *
      ((0 + b * ((0 + c * -2) + b * (0 + x * -2))) + a *
       (((0 + c * (0 + x * -2)) + b * (0 + x * (0 + x * -4))) +
        a * (0 + x * (0 + x * (0 + x * -2)))))))) =
    0 \/
   ~0 + a *
    (0 + a *
     (0 + a *
      (0 + a *
       (((0 + c * -2) + b * (0 + x * -2)) + a *
        (0 + x * (0 + x * -2)))))) =
    0) \/
  ~0 + a *
   (0 + a *
    ((0 + b * (0 + b * 1)) + a *
     (((0 + c * -2) + b * (0 + x * 2)) + a *
      (0 + x * (0 + x * 2))))) =
   0 /\
  0 + a *
  (0 + a *
   (0 + a *
    (0 + a *
     (0 + a *
      (0 + a *
       ((0 + b *
         (0 + b *
          (0 + b *
           ((0 + c * (0 + c * 1)) + b *
            ((0 + c * (0 + x * 2)) + b * (0 + x * (0 + x * 1))))))) +
        a *
        ((0 + b *
          ((0 + c * (0 + c * (0 + c * -4))) + b *
           ((0 + c * (0 + c * (0 + x * -6))) + b *
            ((0 + c * (0 + x * (0 + x * 2))) + b *
             (0 + x * (0 + x * (0 + x * 4))))))) +
         a *
         (((0 + c * (0 + c * (0 + c * (0 + x * -8)))) + b *
           ((0 + c * (0 + c * (0 + x * (0 + x * -24)))) + b *
            ((0 + c * (0 + x * (0 + x * (0 + x * -12)))) + b *
             (0 + x * (0 + x * (0 + x * (0 + x * 5))))))) +
          a *
          (((0 + c * (0 + c * (0 + x * (0 + x * (0 + x * -16))))) +
            b *
            ((0 + c * (0 + x * (0 + x * (0 + x * (0 + x * -20))))) +
             b *
             (0 + x * (0 + x * (0 + x * (0 + x * (0 + x * 2))))))) +
           a *
           (0 + c *
            (0 + x * (0 + x * (0 + x * (0 + x * (0 + x * -8))))))))))))))) =
  0 /\
  ~0 + a *
   (0 + a *
    (0 + a *
     ((0 + b * ((0 + c * 1) + b * (0 + x * 1))) + a *
      (((0 + c * (0 + x * 2)) + b * (0 + x * (0 + x * 3))) + a *
       (0 + x * (0 + x * (0 + x * 2))))))) =
   0) \/
 ~0 + a *
  (0 + a *
   (0 + a *
    (0 + a *
     (((0 + c * (0 + c * 2)) + b *
       ((0 + c * (0 + x * 4)) + b * (0 + x * (0 + x * 2)))) +
      a *
      (((0 + c * (0 + x * (0 + x * 4))) + b *
        (0 + x * (0 + x * (0 + x * 4)))) +
       a * (0 + x * (0 + x * (0 + x * (0 + x * 2))))))))) =
  0 /\
 0 + a *
 (0 + a *
  (0 + a *
   (0 + a *
    (0 + a *
     (0 + a *
      (0 + a *
       (0 + a *
        ((0 + b *
          (0 + b *
           (0 + b *
    ((0 + c * (0 + c * (0 + c * (0 + c * -1)))) + b *
     ((0 + c * (0 + c * (0 + c * (0 + x * -4)))) + b *
      ((0 + c * (0 + c * (0 + x * (0 + x * -6)))) + b *
       ((0 + c * (0 + x * (0 + x * (0 + x * -4)))) + b *
        (0 + x * (0 + x * (0 + x * (0 + x * -1))))))))))) +
 a *
 ((0 + b *
   ((0 + c * (0 + c * (0 + c * (0 + c * (0 + c * 4))))) +
    b *
    ((0 + c * (0 + c * (0 + c * (0 + c * (0 + x * 14))))) +
     b *
     ((0 + c * (0 + c * (0 + c * (0 + x * (0 + x * 12))))) +
      b *
      ((0 + c *
        (0 + c * (0 + x * (0 + x * (0 + x * -8))))) +
       b *
       ((0 + c *
         (0 + x * (0 + x * (0 + x * (0 + x * -16))))) +
        b *
        (0 + x *
         (0 + x * (0 + x * (0 + x * (0 + x * -6))))))))))) +
  a *
  (((0 + c *
     (0 + c * (0 + c * (0 + c * (0 + c * (0 + x * 8)))))) +
    b *
    ((0 + c *
      (0 + c * (0 + c * (0 + c * (0 + x * (0 + x * 48)))))) +
     b *
     ((0 + c *
       (0 + c *
        (0 + c * (0 + x * (0 + x * (0 + x * 88)))))) +
      b *
      ((0 + c *
        (0 + c *
         (0 + x * (0 + x * (0 + x * (0 + x * 50)))))) +
       b *
       ((0 + c *
         (0 + x *
          (0 + x * (0 + x * (0 + x * (0 + x * -12)))))) +
        b *
        (0 + x *
         (0 + x *
          (0 + x * (0 + x * (0 + x * (0 + x * -14))))))))))) +
   a *
   (((0 + c *
      (0 + c *
       (0 + c *
        (0 + c * (0 + x * (0 + x * (0 + x * 32))))))) +
     b *
     ((0 + c *
       (0 + c *
        (0 + c *
         (0 + x * (0 + x * (0 + x * (0 + x * 120))))))) +
      b *
      ((0 + c *
        (0 + c *
         (0 + x *
          (0 + x * (0 + x * (0 + x * (0 + x * 132))))))) +
       b *
       ((0 + c *
         (0 + x *
          (0 + x *
           (0 + x * (0 + x * (0 + x * (0 + x * 28))))))) +
        b *
        (0 + x *
         (0 + x *
          (0 + x *
           (0 + x * (0 + x * (0 + x * (0 + x * -16))))))))))) +
    a *
    (((0 + c *
       (0 + c *
        (0 + c *
         (0 + x *
          (0 + x * (0 + x * (0 + x * (0 + x * 48)))))))) +
      b *
      ((0 + c *
        (0 + c *
         (0 + x *
          (0 + x *
           (0 + x * (0 + x * (0 + x * (0 + x * 112)))))))) +
       b *
       ((0 + c *
         (0 + x *
          (0 + x *
           (0 + x *
            (0 + x * (0 + x * (0 + x * (0 + x * 56)))))))) +
        b *
        (0 + x *
         (0 + x *
          (0 + x *
           (0 + x *
            (0 + x * (0 + x * (0 + x * (0 + x * -9))))))))))) +
     a *
     (((0 + c *
        (0 + c *
         (0 + x *
          (0 + x *
           (0 + x *
            (0 + x * (0 + x * (0 + x * (0 + x * 32))))))))) +
       b *
       ((0 + c *
         (0 + x *
          (0 + x *
           (0 + x *
            (0 + x *
             (0 + x * (0 + x * (0 + x * (0 + x * 36))))))))) +
        b *
        (0 + x *
         (0 + x *
          (0 + x *
                  (0 + x *
                   (0 + x *
                    (0 + x * (0 + x * (0 + x * (0 + x * -2)))))))))))) +
             a *
             (0 + c *
              (0 + x *
               (0 + x *
                (0 + x *
                 (0 + x *
                  (0 + x *
                   (0 + x * (0 + x * (0 + x * (0 + x * 8))))))))))))))))))))))) =
0 /\
0 + a *
(0 + a *
 (0 + a *
  (0 + a *
   (0 + a *
    ((0 + b *
      (0 + b *
       (0 + b *
        ((0 + c * (0 + c * 1)) + b *
         ((0 + c * (0 + x * 2)) + b * (0 + x * (0 + x * 1))))))) +
     a *
     ((0 + b *
       ((0 + c * (0 + c * (0 + c * -4))) + b *
        ((0 + c * (0 + c * (0 + x * -6))) + b *
         ((0 + c * (0 + x * (0 + x * 2))) + b *
          (0 + x * (0 + x * (0 + x * 4))))))) +
      a *
      (((0 + c * (0 + c * (0 + c * (0 + x * -8)))) + b *
        ((0 + c * (0 + c * (0 + x * (0 + x * -24)))) + b *
         ((0 + c * (0 + x * (0 + x * (0 + x * -12)))) + b *
          (0 + x * (0 + x * (0 + x * (0 + x * 5))))))) +
       a *
       (((0 + c * (0 + c * (0 + x * (0 + x * (0 + x * -16))))) +
         b *
         ((0 + c * (0 + x * (0 + x * (0 + x * (0 + x * -20))))) +
          b *
          (0 + x * (0 + x * (0 + x * (0 + x * (0 + x * 2))))))) +
        a *
        (0 + c *
         (0 + x * (0 + x * (0 + x * (0 + x * (0 + x * -8)))))))))))))) =
0 /\
~0 + a *
 (0 + a *
  (0 + a *
   (0 + a *
    ((0 + b *
      ((0 + c * (0 + c * 1)) + b *
       ((0 + c * (0 + x * 2)) + b * (0 + x * (0 + x * 1))))) +
     a *
     (((0 + c * (0 + c * (0 + x * 2))) + b *
       ((0 + c * (0 + x * (0 + x * 6))) + b *
        (0 + x * (0 + x * (0 + x * 4))))) +
      a *
```

```
             (((0 + c * (0 + x * (0 + x * (0 + x * 4)))) + b *
               (0 + x * (0 + x * (0 + x * (0 + x * 5))))) +
              a * (0 + x * (0 + x * (0 + x * (0 + x * (0 + x * 2))))))))))) =
        0)) \/
     0 + a * 1 = 0 /\
     ~0 + b * 1 = 0 /\
     (0 + b * ((0 + c * 1) + b * (0 + x * 1)) = 0 /\
      (0 + c * (0 + c * 1)) + b * (0 + c * (0 + x * 1)) = 0 /\
      (0 + a * 1 = 0 /\
       (0 + b * 1 = 0 /\
        0 + c * 1 = 0 /\ ~(0 + b * 1) + a * (0 + x * 1) = 0 \/
        ~0 + b * 1 = 0 /\
        ~(0 + b * (0 + b * 1)) + a * ((0 + c * -1) + b * (0 + x * 1)) =
         0) \/
       ~0 + a * 1 = 0 /\
       (~0 + a *
         ((0 + b * (0 + b * 1)) + a *
          (((0 + c * -1) + b * (0 + x * 2)) + a * (0 + x * (0 + x * 1)))) =
         0 \/
        ~0 + a * (0 + a * ((0 + b * 1) + a * (0 + x * 2))) = 0)) \/
      ~0 + b * ((0 + c * 1) + b * (0 + x * 1)) = 0 /\
      0 + a *
      ((0 + c * (0 + c * (0 + c * (0 + c * 1)))) + b *
       ((0 + c * (0 + c * (0 + c * (0 + x * 2)))) + b *
        (0 + c * (0 + c * (0 + x * (0 + x * 1)))))) =
      0 /\
      ~(0 + b * (0 + b * ((0 + c * 1) + b * (0 + x * 1)))) + a *
       ((0 + c * (0 + c * -1)) + b * (0 + b * (0 + x * (0 + x * 1)))) =
       0) \/
     0 + a * 1 = 0 /\
     0 + b * 1 = 0 /\
     ~0 + c * 1 = 0 /\
     (0 + a * 1 = 0 /\
      (0 + b * 1 = 0 /\
       0 + c * 1 = 0 /\ ~(0 + b * 1) + a * (0 + x * 1) = 0 \/
       ~0 + b * 1 = 0 /\
       ~(0 + b * (0 + b * 1)) + a * ((0 + c * -1) + b * (0 + x * 1)) =
        0) \/
      ~0 + a * 1 = 0 /\
      (~0 + a *
        ((0 + b * (0 + b * 1)) + a *
         (((0 + c * -1) + b * (0 + x * 2)) + a * (0 + x * (0 + x * 1)))) =
        0 \/
       ~0 + a * (0 + a * ((0 + b * 1) + a * (0 + x * 2))) = 0))) /\
    ((0 + c * 1) + b * (0 + x * 1)) + a * (0 + x * (0 + x * 1)) = 0)>>
// complex.p042 - long running for F#
CPU time (user): 0.0680050000001
- : fol formula = <<true>>
// complex.p043 - long running for F#
CPU time (user): 0.16001
- : fol formula = <<true>>
// complex.p044 - not run
// complex.p045 - not run
// complex.p046 - not run
// complex.p047
```

```
CPU time (user): 7.19645
- : fol formula = <<true>>
// complex.p048
CPU time (user): 0.00400000000002
- : fol formula =
<<~(((0 + y * 1) + x * -1 = 0 /\
     ((0 + y * (0 + y * 1)) + x * (0 + x * -1) = 0 \/
      ~(0 + y * (0 + y * 1)) + x * (0 + x * -1) = 0)) /\
    ~(0 + y * -1) + x * 1 = 0)>>
// complex.p049
CPU time (user): 0.00400000000002
- : fol formula =
<<~(((0 + y * 2) + x * -2 = 0 /\
     ((0 + y * (0 + y * 2)) + x * (0 + x * -2) = 0 \/
      ~(0 + y * (0 + y * 2)) + x * (0 + x * -2) = 0)) /\
    ~(0 + y * -1) + x * 1 = 0)>>
// complex.p050
CPU time (user): 0.00400100000002
- : fol formula = <<true>>
// complex.p051
CPU time (user): 0.
- : fol formula = <<true>>
// complex.p054
CPU time (user): 0.276017
- : fol formula = <<true>>
// complex.p055
CPU time (user): 0.204013
- : fol formula = <<true>>
// complex.p056
CPU time (user): 0.0680040000001
- : fol formula = <<true>>
// complex.p057 - Not run - true
// complex.p058 - Not run - Large formula
// complex.p059 - Long running
// complex.p060
CPU time (user): 0.0200020000001
- : fol formula =
<<~((0 + a * 1 = 0 /\
     (0 + b * 1 = 0 /\ 0 + c * 1 = 0 /\ ~0 + a * (0 + x * 1) = 0 \/
      ~0 + b * 1 = 0 /\
      ~(0 + b * ((0 + c * (0 + c * -1)) + b * (0 + c * (0 + x * -1)))) +
       a *
       ((0 + c * (0 + c * (0 + x * -1))) + b *
        (0 + c * (0 + x * (0 + x * -1)))) =
       0) \/
     ~0 + a * 1 = 0 /\
     (~0 + a *
       (0 + a *
        (((0 + c * (0 + c * (0 + c * -1))) + b *
          ((0 + c * (0 + c * (0 + x * -2))) + b *
           (0 + c * (0 + x * (0 + x * -1))))) +
         a *
         (((0 + c * (0 + c * (0 + x * (0 + x * -2)))) + b *
           (0 + c * (0 + x * (0 + x * (0 + x * -2))))) +
          a * (0 + c * (0 + x * (0 + x * (0 + x * (0 + x * -1)))))))) =
       0 \/
      ~0 + a *
       (0 + a *
        ((0 + b *
          ((0 + c * (0 + c * -1)) + b *
           ((0 + c * (0 + x * -2)) + b * (0 + x * (0 + x * -1))))) +
         a *
         ((0 + b *
           ((0 + c * (0 + x * (0 + x * -2))) + b *
            (0 + x * (0 + x * (0 + x * -2))))) +
          a * (0 + b * (0 + x * (0 + x * (0 + x * (0 + x * -1)))))))) =
       0)) /\
    ((0 + c * 1) + b * (0 + x * 1)) + a * (0 + x * (0 + x * 1)) = 0 \/
    (0 + a * 1 = 0 /\
     (0 + b * 1 = 0 /\
      0 + c * 1 = 0 /\ ~(0 + b * 1) + a * (0 + x * 1) = 0 \/
      ~0 + b * 1 = 0 /\
      ~(0 + b * (0 + b * ((0 + c * 1) + b * (0 + x * 1)))) + a *
       ((0 + c * (0 + c * -1)) + b * (0 + b * (0 + x * (0 + x * 1)))) =
       0) \/
     ~0 + a * 1 = 0 /\
     ~0 + a *
      (0 + a *
       (0 + a *
        (((0 + c * (0 + c * 1)) + b *
          ((0 + c * (0 + x * 2)) + b * (0 + x * (0 + x * 1)))) +
         a *
         (((0 + c * (0 + x * (0 + x * 2))) + b *
           (0 + x * (0 + x * (0 + x * 2)))) +
          a * (0 + x * (0 + x * (0 + x * (0 + x * 1)))))))) =
      0) /\
    ((0 + c * 1) + b * (0 + x * 1)) + a * (0 + x * (0 + x * 1)) = 0)>>
// complex.p061
CPU time (user): 0.192012
- : fol formula = <<true>>
// complex.p062 - Not run
// complex.p063
CPU time (user): 0.00800100000004
- : fol formula = <<true>>
// complex.p064
CPU time (user): 0.020001
- : fol formula = <<true>>
// complex.p065
CPU time (user): 1.684105
- : fol formula = <<true>>
// complex.p066 - Long running
// complex.p067
CPU time (user): 0.
- : fol formula = <<true>>
// complex.p068
CPU time (user): 0.
- : fol formula = <<true>>
// complex.p069
CPU time (user): 0.0119999999999
- : fol formula = <<true>>
```

```
// complex.p070
CPU time (user): 0.024002
- : fol formula = <<true>>
// complex.p071
CPU time (user): 0.076005
- : fol formula = <<0 + x * 1 = 0 /\ ~0 + y * 1 = 0 \/ ~0 + x * 1 = 0>>
// complex.p072
CPU time (user): 0.
- : fol formula =
<<~(0 + x * 1 = 0 /\
    (0 + x * 2 = 0 /\ 0 + y * 1 = 0 /\ ~0 + z * 1 = 0 \/
     ~0 + x * 2 = 0 /\
     ~0 + x * ((0 + y * (0 + y * -1)) + x * (0 + z * 4)) = 0))>>
// complex.p073
CPU time (user): 0.216013
- : fol formula = <<true>>
// complex.p074
CPU time (user): 0.0119999999999
- : fol formula = <<true>>
// complex.p075
CPU time (user): 0.
- : fol formula =
<<(~0 + b * 1 = 0 /\
   ((0 + b * (0 + c * 1)) + a * (0 + d * -1) = 0 /\ 0 + d * 1 = 0 \/
    ~(0 + b * (0 + c * 1)) + a * (0 + d * -1) = 0) \/
   0 + b * 1 = 0 /\ 0 + d * 1 = 0 /\ ~0 + a * 1 = 0 /\ 0 + c * 1 = 0 \/
   0 + b * 1 = 0 /\ ~0 + d * 1 = 0 /\ ~0 + a * 1 = 0) /\
  (~0 + d * 1 = 0 /\
   ((0 + b * (0 + c * -1)) + a * (0 + d * 1) = 0 /\ 0 + b * 1 = 0 \/
    ~(0 + b * (0 + c * -1)) + a * (0 + d * 1) = 0) \/
   0 + b * 1 = 0 /\ 0 + d * 1 = 0 /\ ~0 + c * 1 = 0 /\ 0 + a * 1 = 0 \/
   0 + d * 1 = 0 /\ ~0 + b * 1 = 0 /\ ~0 + c * 1 = 0)>>
// complex.p076
CPU time (user): 0.0360020000001
- : fol formula = <<true>>
// complex.p077
CPU time (user): 4.096256
- : fol formula = <<false>>
// complex.p078
CPU time (user): 0.068004
- : fol formula = <<true>>
// complex.p079
CPU time (user): 0.00800000000004
- : fol formula = <<true>>
// complex.p080
CPU time (user): 0.0120009999999
- : fol formula = <<true>>
// complex.p081
CPU time (user): 0.00800100000004
- : fol formula = <<true>>
// complex.p082
CPU time (user): 0.00400000000002
- : fol formula = <<true>>
// complex.p083
CPU time (user): 0.180011
```

```
- : term = <<|0|>>
// complex.p084
CPU time (user): 0.00400000000002
- : fol formula =
<<~(((0 + u * -1) + d * 1 = 0 /\
     (0 + v * 1) + b * 1 = 0 /\
     ((0 + u * -1) + a * 1 = 0 /\
      ((0 + v * -1) + c * 1 = 0 \/ ~(0 + v * -1) + c * 1 = 0) \/
      ~(0 + u * -1) + a * 1 = 0) \/
     (0 + v * 1) + b * 1 = 0 /\
     ~(0 + u * -1) + d * 1 = 0 /\
     ((0 + u * -1) + a * 1 = 0 \/ ~(0 + u * -1) + a * 1 = 0) \/
     ~(0 + v * 1) + b * 1 = 0 /\
     ((((((0 + v * (0 + v * -1)) + u * (0 + u * -1)) + d * (0 + u * 1)) +
        c * (0 + v * 1)) +
       b * ((0 + v * -1) + c * 1)) +
      a * ((0 + u * 1) + d * -1) = 0 \/
      ~(((((0 + v * (0 + v * -1)) + u * (0 + u * -1)) + d * (0 + u * 1)) +
         c * (0 + v * 1)) +
        b * ((0 + v * -1) + c * 1)) +
       a * ((0 + u * 1) + d * -1) = 0)) /\
    ~(0 + d * -1) + a * 1 = 0)>>
// complex.p085
CPU time (user): 0.00399999999991
- : fol formula = <<true>>
// complex.p086
CPU time (user): 0.00400000000002
- : fol formula = <<true>>
// complex.p087
- : fol formula = <<false>>
- : dummy_interactive = END_INTERACTIVE
#
```

real.ml

```
#use "real.ml";;
```

```
# #use "real.ml";;
val poly_diffn : term -> int -> term -> term = <fun>
val poly_diff : string list -> term -> term = <fun>
val rel_signs : (string * sign list) list =
  [("=", [Zero]); ("<=", [Zero; Negative]); (">=", [Zero; Positive]);
   ("<", [Negative]); (">", [Positive])]
File "/home/student/Harrison/OCaml/real.ml", line 32, characters 11-69:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
R (_, [])
val testform : (term * sign) list -> fol formula -> bool = <fun>
val inferpsign : sign list * sign list -> sign list = <fun>
val condense : sign list list -> sign list list = <fun>
val inferisign : sign list list -> sign list list = <fun>
val dedmatrix : (sign list list -> 'a) -> sign list list -> 'a = <fun>
val pdivide_pos :
  string list -> (term * sign) list -> term -> term -> term = <fun>
val split_sign :
  (term * sign) list ->
  term -> ((term * sign) list -> fol formula) -> fol formula = <fun>
val split_trichotomy :
  (term * sign) list ->
  term ->
  ((term * sign) list -> fol formula) ->
  ((term * sign) list -> fol formula) -> fol formula = <fun>
val casesplit :
  string list ->
  term list ->
  term list ->
  (sign list list -> fol formula) -> (term * sign) list -> fol formula =
  <fun>
val delconst :
  string list ->
  term list ->
  term ->
  term list ->
  (sign list list -> fol formula) -> (term * sign) list -> fol formula =
  <fun>
val matrix :
  string list ->
  term list ->
  (sign list list -> fol formula) -> (term * sign) list -> fol formula =
  <fun>
File "/home/student/Harrison/OCaml/real.ml", line 137, characters 26-276:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
(Forall (_, _)|Iff (_, _)|Imp (_, _)|Or (_, _)|And (_, _)|Not _|Atom _|True|
False)
val basic_real_qelim : string list -> fol formula -> fol formula =
  <fun>
val real_qelim : fol formula -> fol formula = <fun>
- : dummy_interactive = START_INTERACTIVE
// real.p001
- : fol formula = <<false>>
// real.p002
- : fol formula = <<true>>
// real.p003
- : fol formula = <<false>>
// real.p004
testform is now traced.
testform <--
  [(<<|-3 + x * 2|>>, Negative);
   (<<|2 + x * (-3 + x * 1)|>>, Positive)]
testform --> <fun>
testform* <-- <<2 + x * (-3 + x * 1) = 0 /\ -3 + x * 2 = 0>>
testform* --> false
testform <--
  [(<<|-3 + x * 2|>>, Negative); (<<|2 + x * (-3 + x * 1)|>>, Zero)]
testform --> <fun>
testform* <-- <<2 + x * (-3 + x * 1) = 0 /\ -3 + x * 2 = 0>>
testform* --> false
testform <--
  [(<<|-3 + x * 2|>>, Negative);
   (<<|2 + x * (-3 + x * 1)|>>, Negative)]
testform --> <fun>
testform* <-- <<2 + x * (-3 + x * 1) = 0 /\ -3 + x * 2 = 0>>
testform* --> false
testform <--
  [(<<|-3 + x * 2|>>, Zero); (<<|2 + x * (-3 + x * 1)|>>, Negative)]
testform --> <fun>
testform* <-- <<2 + x * (-3 + x * 1) = 0 /\ -3 + x * 2 = 0>>
testform* --> false
testform <--
  [(<<|-3 + x * 2|>>, Positive);
   (<<|2 + x * (-3 + x * 1)|>>, Negative)]
testform --> <fun>
testform* <-- <<2 + x * (-3 + x * 1) = 0 /\ -3 + x * 2 = 0>>
testform* --> false
testform <--
  [(<<|-3 + x * 2|>>, Positive); (<<|2 + x * (-3 + x * 1)|>>, Zero)]
testform --> <fun>
testform* <-- <<2 + x * (-3 + x * 1) = 0 /\ -3 + x * 2 = 0>>
testform* --> false
testform <--
  [(<<|-3 + x * 2|>>, Positive);
   (<<|2 + x * (-3 + x * 1)|>>, Positive)]
testform --> <fun>
testform* <-- <<2 + x * (-3 + x * 1) = 0 /\ -3 + x * 2 = 0>>
testform* --> false
- : fol formula = <<false>>
testform is no longer traced.
// real.p005
```

```
- : fol formula = <<true>>
// real.p006
- : fol formula =
<<0 + a * 1 = 0 /\
  (0 + b * 1 = 0 /\ 0 + c * 1 = 0 \/
   ~0 + b * 1 = 0 /\ (0 + b * 1 > 0 \/ ~0 + b * 1 > 0)) \/
  ~0 + a * 1 = 0 /\
  (0 + a * 1 > 0 /\
   (0 + a * ((0 + b * (0 + b * -1)) + a * (0 + c * 4)) = 0 \/
    ~0 + a * ((0 + b * (0 + b * -1)) + a * (0 + c * 4)) = 0 /\
    ~0 + a * ((0 + b * (0 + b * -1)) + a * (0 + c * 4)) > 0) \/
   ~0 + a * 1 > 0 /\
   (0 + a * ((0 + b * (0 + b * -1)) + a * (0 + c * 4)) = 0 \/
    ~0 + a * ((0 + b * (0 + b * -1)) + a * (0 + c * 4)) = 0 /\ 0 + a *
    ((0 + b * (0 + b * -1)) + a * (0 + c * 4)) > 0))>>
// real.p007
- : fol formula = <<false>>
// real.p008
- : fol formula = <<true>>
// real.p009
- : fol formula = <<true>>
- : dummy_interactive = END_INTERACTIVE
File "/home/student/Harrison/OCaml/real.ml", line 185, characters 2-243:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
Fn ("", _)
val grpterm : term -> term = <fun>
File "/home/student/Harrison/OCaml/real.ml", line 192, characters 12-154:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
(Exists (_, _)|Forall (_, _)|Iff (_, _)|Imp (_, _)|Or (_, _)|And (_, _)|
Not _|True|False)
val grpform : fol formula -> fol formula = <fun>
- : dummy_interactive = START_INTERACTIVE
// real.p010
4 equations and 8 pending critical pairs + 0 deferred
5 equations and 12 pending critical pairs + 0 deferred
6 equations and 16 pending critical pairs + 0 deferred
7 equations and 27 pending critical pairs + 0 deferred
8 equations and 51 pending critical pairs + 0 deferred
9 equations and 70 pending critical pairs + 0 deferred
10 equations and 81 pending critical pairs + 0 deferred
11 equations and 78 pending critical pairs + 0 deferred
12 equations and 85 pending critical pairs + 0 deferred
13 equations and 114 pending critical pairs + 0 deferred
14 equations and 151 pending critical pairs + 0 deferred
15 equations and 180 pending critical pairs + 0 deferred
16 equations and 247 pending critical pairs + 0 deferred
17 equations and 298 pending critical pairs + 0 deferred
18 equations and 356 pending critical pairs + 0 deferred
19 equations and 404 pending critical pairs + 0 deferred
20 equations and 485 pending critical pairs + 0 deferred
21 equations and 530 pending critical pairs + 0 deferred
22 equations and 583 pending critical pairs + 0 deferred
23 equations and 642 pending critical pairs + 0 deferred
```

```
24 equations and 730 pending critical pairs + 0 deferred
25 equations and 779 pending critical pairs + 0 deferred
26 equations and 794 pending critical pairs + 0 deferred
27 equations and 819 pending critical pairs + 1 deferred
28 equations and 918 pending critical pairs + 1 deferred
29 equations and 901 pending critical pairs + 1 deferred
30 equations and 1005 pending critical pairs + 1 deferred
31 equations and 1086 pending critical pairs + 1 deferred
32 equations and 1155 pending critical pairs + 1 deferred
32 equations and 1000 pending critical pairs + 1 deferred
32 equations and 0 pending critical pairs + 1 deferred
32 equations and 0 pending critical pairs + 0 deferred
val eqs : fol formula list =
  [<<i(x4 * x5) = i(x5) * i(x4)>>; <<i(i(x1)) = x1>>; <<i(1) = 1>>;
   <<x0 * i(x0) = 1>>; <<x0 * i(x0) * x3 = x3>>; <<x1 * 1 = x1>>;
   <<i(x1) * x1 * x2 = x2>>; <<1 * x = x>>; <<i(x) * x = 1>>;
   <<(x * y) * z = x * y * z>>]
// real.p011
val fm : fol formula =
  <<(forall x4.
       x4 > 1 ==>
       (forall x5.
          x5 > 1 ==> (x4 * (1 + 2 * x5))^2 > x5^2 * (1 + 2 * x4^2))) /\
    (forall x1. x1 > 1 ==> x1^2^2 > x1) /\
    2^2 > 2 /\
    (forall x0. x0 > 1 ==> x0 * (1 + 2 * x0^2) > 2) /\
    (forall x0.
       x0 > 1 ==>
       (forall x3. x3 > 1 ==> x0 * (1 + 2 * x0^2 * (1 + 2 * x3)) > x3)) /\
    (forall x1. x1 > 1 ==> x1 * (1 + 2 * 2) > x1) /\
    (forall x1.
       x1 > 1 ==>
       (forall x2. x2 > 1 ==> x1^2 * (1 + 2 * x1 * (1 + 2 * x2)) > x2)) /\
    (forall x. x > 1 ==> 2 * (1 + 2 * x) > x) /\
    (forall x. x > 1 ==> x^2 * (1 + 2 * x) > 2) /\
    (forall x.
       x > 1 ==>
       (forall y.
          y > 1 ==>
          (forall z.
             z > 1 ==> (x * (1 + 2 * y)) * (1 + 2 * z) > x *
             (1 + 2 * y * (1 + 2 * z)))))>>
// real.p012
- : fol formula = <<true>>
- : dummy_interactive = END_INTERACTIVE
val real_qelim' : fol formula -> fol formula = <fun>
- : fol formula = <<true>>
- : dummy_interactive = START_INTERACTIVE
val casesplit :
  string list ->
  term list ->
  term list ->
  (sign list list -> fol formula) -> (term * sign) list -> fol formula =
  <fun>
val delconst :
  string list ->
  term list ->
  term ->
  term list ->
  (sign list list -> fol formula) -> (term * sign) list -> fol formula =
  <fun>
val matrix :
  string list ->
  term list ->
  (sign list list -> fol formula) -> (term * sign) list -> fol formula =
  <fun>
val monicize :
  string list ->
  term list ->
  (sign list list -> fol formula) -> (term * sign) list -> fol formula =
  <fun>
File "/home/student/Harrison/OCaml/real.ml", line 257, characters 26-276:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
(Forall (_, _)|Iff (_, _)|Imp (_, _)|Or (_, _)|And (_, _)|Not _|Atom _|True|
False)
val basic_real_qelim : string list -> fol formula -> fol formula =
  <fun>
val real_qelim : fol formula -> fol formula = <fun>
val real_qelim' : fol formula -> fol formula = <fun>
- : dummy_interactive = END_INTERACTIVE
#
```

grobner.ml

```
#use grobner ml";;
```

```
# #use "grobner.ml";;
val mmul : num * int list -> num * int list -> num * int list = <fun>
val mdiv : num * int list -> num * int list -> num * int list = <fun>
val mlcm : 'a * 'b list -> 'c * 'b list -> num * 'b list = <fun>
val morder_lt : int list -> int list -> bool = <fun>
val mpoly_mmul :
  num * int list -> (num * int list) list -> (num * int list) list =
  <fun>
val mpoly_neg : (num * '_a) list -> (num * '_a) list = <fun>
val mpoly_const : 'a list -> num -> (num * int list) list = <fun>
val mpoly_var : 'a list -> 'a -> (num * int list) list = <fun>
val mpoly_add :
  (num * int list) list ->
  (num * int list) list -> (num * int list) list = <fun>
val mpoly_sub :
  (num * int list) list ->
  (num * int list) list -> (num * int list) list = <fun>
val mpoly_mul :
  (num * int list) list ->
  (num * int list) list -> (num * int list) list = <fun>
val mpoly_pow :
  'a list -> (num * int list) list -> int -> (num * int list) list =
  <fun>
val mpoly_inv : (num * int list) list -> (num * int list) list = <fun>
val mpoly_div :
  (num * int list) list ->
  (num * int list) list -> (num * int list) list = <fun>
val mpolynate : string list -> term -> (num * int list) list = <fun>
val mpolyatom : string list -> fol formula -> (num * int list) list =
  <fun>
val reduce1 :
  num * int list -> (num * int list) list -> (num * int list) list =
  <fun>
val reduceb :
  num * int list -> (num * int list) list list -> (num * int list) list =
  <fun>
val reduce :
  (num * int list) list list ->
  (num * int list) list -> (num * int list) list = <fun>
val spoly :
  (num * int list) list ->
  (num * int list) list -> (num * int list) list = <fun>
val grobner :
  (num * int list) list list ->
  ((num * int list) list * (num * int list) list) list ->
  (num * int list) list list = <fun>
val groebner : (num * int list) list list -> (num * int list) list list =
  <fun>
val rabinowitsch :
  'a list -> 'a -> (num * int list) list -> (num * int list) list =
  <fun>
val grobner_trivial : fol formula list -> bool = <fun>
val grobner_decide : fol formula -> bool = <fun>
- : dummy_interactive = START_INTERACTIVE
// grobner.p001
3 basis elements and 3 pairs
3 basis elements and 2 pairs
- : bool = true
// grobner.p002
3 basis elements and 3 pairs
3 basis elements and 2 pairs
4 basis elements and 4 pairs
4 basis elements and 3 pairs
4 basis elements and 2 pairs
4 basis elements and 1 pairs
4 basis elements and 0 pairs
- : bool = false
// grobner.p003
4 basis elements and 6 pairs
5 basis elements and 9 pairs
5 basis elements and 8 pairs
6 basis elements and 12 pairs
7 basis elements and 17 pairs
8 basis elements and 23 pairs
9 basis elements and 30 pairs
9 basis elements and 29 pairs
9 basis elements and 28 pairs
10 basis elements and 36 pairs
11 basis elements and 45 pairs
12 basis elements and 55 pairs
13 basis elements and 66 pairs
13 basis elements and 65 pairs
14 basis elements and 77 pairs
15 basis elements and 90 pairs
16 basis elements and 104 pairs
16 basis elements and 103 pairs
17 basis elements and 118 pairs
18 basis elements and 134 pairs
19 basis elements and 151 pairs
4 basis elements and 6 pairs
5 basis elements and 9 pairs
5 basis elements and 8 pairs
6 basis elements and 12 pairs
7 basis elements and 17 pairs
8 basis elements and 23 pairs
9 basis elements and 30 pairs
10 basis elements and 38 pairs
11 basis elements and 47 pairs
12 basis elements and 57 pairs
13 basis elements and 68 pairs
14 basis elements and 80 pairs
15 basis elements and 93 pairs
15 basis elements and 92 pairs
16 basis elements and 106 pairs
```

```
17 basis elements and 121 pairs
18 basis elements and 137 pairs
18 basis elements and 136 pairs
19 basis elements and 153 pairs
20 basis elements and 171 pairs
21 basis elements and 190 pairs
- : bool = true
// grobner.p004
4 basis elements and 6 pairs
5 basis elements and 9 pairs
5 basis elements and 8 pairs
6 basis elements and 12 pairs
7 basis elements and 17 pairs
8 basis elements and 23 pairs
9 basis elements and 30 pairs
9 basis elements and 29 pairs
9 basis elements and 28 pairs
10 basis elements and 36 pairs
11 basis elements and 45 pairs
12 basis elements and 55 pairs
13 basis elements and 66 pairs
13 basis elements and 65 pairs
14 basis elements and 77 pairs
15 basis elements and 90 pairs
16 basis elements and 104 pairs
16 basis elements and 103 pairs
17 basis elements and 118 pairs
18 basis elements and 134 pairs
19 basis elements and 151 pairs
4 basis elements and 6 pairs
5 basis elements and 9 pairs
5 basis elements and 8 pairs
6 basis elements and 12 pairs
7 basis elements and 17 pairs
8 basis elements and 23 pairs
9 basis elements and 30 pairs
10 basis elements and 38 pairs
11 basis elements and 47 pairs
12 basis elements and 57 pairs
13 basis elements and 68 pairs
14 basis elements and 80 pairs
15 basis elements and 93 pairs
15 basis elements and 92 pairs
16 basis elements and 106 pairs
17 basis elements and 121 pairs
18 basis elements and 137 pairs
18 basis elements and 136 pairs
19 basis elements and 153 pairs
20 basis elements and 171 pairs
21 basis elements and 190 pairs
CPU time (user): 0.00399999999991
CPU time (user): 0.200013
- : fol formula * bool = (<<true>>, true)
// grobner.p005
3 basis elements and 3 pairs
```

```
3 basis elements and 2 pairs
CPU time (user): 0.
- : bool = true
// grobner.p006
3 basis elements and 3 pairs
3 basis elements and 2 pairs
4 basis elements and 4 pairs
4 basis elements and 3 pairs
4 basis elements and 2 pairs
4 basis elements and 1 pairs
4 basis elements and 0 pairs
CPU time (user): 0.
- : bool = false
// grobner.p007
4 basis elements and 6 pairs
5 basis elements and 9 pairs
5 basis elements and 8 pairs
6 basis elements and 12 pairs
7 basis elements and 17 pairs
8 basis elements and 23 pairs
9 basis elements and 30 pairs
9 basis elements and 29 pairs
9 basis elements and 28 pairs
10 basis elements and 36 pairs
11 basis elements and 45 pairs
12 basis elements and 55 pairs
13 basis elements and 66 pairs
13 basis elements and 65 pairs
14 basis elements and 77 pairs
15 basis elements and 90 pairs
16 basis elements and 104 pairs
16 basis elements and 103 pairs
17 basis elements and 118 pairs
18 basis elements and 134 pairs
19 basis elements and 151 pairs
4 basis elements and 6 pairs
5 basis elements and 9 pairs
5 basis elements and 8 pairs
6 basis elements and 12 pairs
7 basis elements and 17 pairs
8 basis elements and 23 pairs
9 basis elements and 30 pairs
10 basis elements and 38 pairs
11 basis elements and 47 pairs
12 basis elements and 57 pairs
13 basis elements and 68 pairs
14 basis elements and 80 pairs
15 basis elements and 93 pairs
15 basis elements and 92 pairs
16 basis elements and 106 pairs
17 basis elements and 121 pairs
18 basis elements and 137 pairs
18 basis elements and 136 pairs
19 basis elements and 153 pairs
20 basis elements and 171 pairs
```

```
21 basis elements and 190 pairs
CPU time (user): 0.00400000000002
- : bool = true
// grobner.p008
4 basis elements and 6 pairs
4 basis elements and 5 pairs
5 basis elements and 8 pairs
6 basis elements and 12 pairs
7 basis elements and 17 pairs
8 basis elements and 23 pairs
8 basis elements and 22 pairs
8 basis elements and 21 pairs
8 basis elements and 20 pairs
8 basis elements and 19 pairs
8 basis elements and 18 pairs
8 basis elements and 17 pairs
8 basis elements and 16 pairs
8 basis elements and 15 pairs
8 basis elements and 14 pairs
8 basis elements and 13 pairs
8 basis elements and 12 pairs
8 basis elements and 11 pairs
8 basis elements and 10 pairs
8 basis elements and 9 pairs
8 basis elements and 8 pairs
CPU time (user): 0.
- : bool = true
// grobner.p009
7 basis elements and 21 pairs
7 basis elements and 20 pairs
7 basis elements and 19 pairs
8 basis elements and 25 pairs
8 basis elements and 24 pairs
8 basis elements and 23 pairs
8 basis elements and 22 pairs
9 basis elements and 29 pairs
9 basis elements and 28 pairs
9 basis elements and 27 pairs
10 basis elements and 35 pairs
11 basis elements and 44 pairs
11 basis elements and 43 pairs
12 basis elements and 53 pairs
13 basis elements and 64 pairs
14 basis elements and 76 pairs
15 basis elements and 89 pairs
15 basis elements and 88 pairs
16 basis elements and 102 pairs
CPU time (user): 0.00400000000002
- : bool = true
// grobner.p010
7 basis elements and 21 pairs
7 basis elements and 20 pairs
7 basis elements and 19 pairs
8 basis elements and 25 pairs
9 basis elements and 32 pairs
```

```
9 basis elements and 31 pairs
9 basis elements and 30 pairs
10 basis elements and 38 pairs
11 basis elements and 47 pairs
11 basis elements and 46 pairs
12 basis elements and 56 pairs
13 basis elements and 67 pairs
13 basis elements and 66 pairs
14 basis elements and 78 pairs
15 basis elements and 91 pairs
16 basis elements and 105 pairs
17 basis elements and 120 pairs
17 basis elements and 119 pairs
18 basis elements and 135 pairs
CPU time (user): 0.00400100000002
- : bool = true
// grobner.p011
3 basis elements and 3 pairs
4 basis elements and 5 pairs
5 basis elements and 8 pairs
6 basis elements and 12 pairs
6 basis elements and 11 pairs
7 basis elements and 16 pairs
8 basis elements and 22 pairs
9 basis elements and 29 pairs
9 basis elements and 28 pairs
9 basis elements and 27 pairs
10 basis elements and 35 pairs
10 basis elements and 34 pairs
10 basis elements and 33 pairs
10 basis elements and 32 pairs
10 basis elements and 31 pairs
11 basis elements and 40 pairs
11 basis elements and 39 pairs
11 basis elements and 38 pairs
11 basis elements and 37 pairs
11 basis elements and 36 pairs
11 basis elements and 35 pairs
CPU time (user): 0.
- : bool = true
// grobner.p012
3 basis elements and 3 pairs
4 basis elements and 5 pairs
5 basis elements and 8 pairs
6 basis elements and 12 pairs
6 basis elements and 11 pairs
7 basis elements and 16 pairs
7 basis elements and 15 pairs
8 basis elements and 21 pairs
9 basis elements and 28 pairs
10 basis elements and 36 pairs
11 basis elements and 45 pairs
12 basis elements and 55 pairs
13 basis elements and 66 pairs
13 basis elements and 65 pairs
```

```
14 basis elements and 77 pairs
15 basis elements and 90 pairs
16 basis elements and 104 pairs
16 basis elements and 103 pairs
16 basis elements and 102 pairs
16 basis elements and 101 pairs
16 basis elements and 100 pairs
CPU time (user): 0.00400000000002
- : bool = true
// grobner.p013
3 basis elements and 3 pairs
4 basis elements and 5 pairs
5 basis elements and 8 pairs
6 basis elements and 12 pairs
7 basis elements and 17 pairs
8 basis elements and 23 pairs
9 basis elements and 30 pairs
10 basis elements and 38 pairs
11 basis elements and 47 pairs
12 basis elements and 57 pairs
13 basis elements and 68 pairs
14 basis elements and 80 pairs
15 basis elements and 93 pairs
16 basis elements and 107 pairs
17 basis elements and 122 pairs
18 basis elements and 138 pairs
19 basis elements and 155 pairs
20 basis elements and 173 pairs
20 basis elements and 172 pairs
21 basis elements and 191 pairs
21 basis elements and 190 pairs
22 basis elements and 210 pairs
23 basis elements and 231 pairs
24 basis elements and 253 pairs
24 basis elements and 252 pairs
25 basis elements and 275 pairs
25 basis elements and 274 pairs
25 basis elements and 273 pairs
26 basis elements and 297 pairs
27 basis elements and 322 pairs
27 basis elements and 321 pairs
27 basis elements and 320 pairs
27 basis elements and 319 pairs
27 basis elements and 318 pairs
27 basis elements and 317 pairs
28 basis elements and 343 pairs
29 basis elements and 370 pairs
29 basis elements and 369 pairs
29 basis elements and 368 pairs
29 basis elements and 367 pairs
29 basis elements and 366 pairs
29 basis elements and 365 pairs
30 basis elements and 393 pairs
31 basis elements and 422 pairs
32 basis elements and 452 pairs
```

```
32 basis elements and 451 pairs
32 basis elements and 450 pairs
32 basis elements and 449 pairs
33 basis elements and 480 pairs
33 basis elements and 479 pairs
33 basis elements and 478 pairs
34 basis elements and 510 pairs
34 basis elements and 509 pairs
35 basis elements and 542 pairs
35 basis elements and 541 pairs
35 basis elements and 540 pairs
35 basis elements and 539 pairs
36 basis elements and 573 pairs
36 basis elements and 572 pairs
36 basis elements and 571 pairs
36 basis elements and 570 pairs
36 basis elements and 569 pairs
36 basis elements and 568 pairs
36 basis elements and 567 pairs
37 basis elements and 602 pairs
37 basis elements and 601 pairs
37 basis elements and 600 pairs
37 basis elements and 599 pairs
37 basis elements and 598 pairs
37 basis elements and 597 pairs
37 basis elements and 596 pairs
37 basis elements and 595 pairs
37 basis elements and 594 pairs
37 basis elements and 593 pairs
37 basis elements and 592 pairs
37 basis elements and 591 pairs
37 basis elements and 590 pairs
37 basis elements and 589 pairs
38 basis elements and 625 pairs
38 basis elements and 624 pairs
38 basis elements and 623 pairs
38 basis elements and 622 pairs
38 basis elements and 621 pairs
38 basis elements and 620 pairs
38 basis elements and 619 pairs
38 basis elements and 618 pairs
38 basis elements and 617 pairs
38 basis elements and 616 pairs
38 basis elements and 615 pairs
38 basis elements and 614 pairs
38 basis elements and 613 pairs
38 basis elements and 612 pairs
38 basis elements and 611 pairs
38 basis elements and 610 pairs
38 basis elements and 609 pairs
38 basis elements and 608 pairs
38 basis elements and 607 pairs
38 basis elements and 606 pairs
38 basis elements and 605 pairs
38 basis elements and 604 pairs
```

```
38 basis elements and 603 pairs
38 basis elements and 602 pairs
38 basis elements and 601 pairs
38 basis elements and 600 pairs
38 basis elements and 599 pairs
38 basis elements and 598 pairs
38 basis elements and 597 pairs
38 basis elements and 596 pairs
38 basis elements and 595 pairs
38 basis elements and 594 pairs
38 basis elements and 593 pairs
38 basis elements and 592 pairs
38 basis elements and 591 pairs
38 basis elements and 590 pairs
38 basis elements and 589 pairs
38 basis elements and 588 pairs
38 basis elements and 587 pairs
38 basis elements and 586 pairs
38 basis elements and 585 pairs
38 basis elements and 584 pairs
38 basis elements and 583 pairs
38 basis elements and 582 pairs
38 basis elements and 581 pairs
38 basis elements and 580 pairs
38 basis elements and 579 pairs
38 basis elements and 578 pairs
38 basis elements and 577 pairs
38 basis elements and 576 pairs
38 basis elements and 575 pairs
38 basis elements and 574 pairs
38 basis elements and 573 pairs
38 basis elements and 572 pairs
38 basis elements and 571 pairs
38 basis elements and 570 pairs
38 basis elements and 569 pairs
38 basis elements and 568 pairs
38 basis elements and 567 pairs
38 basis elements and 566 pairs
38 basis elements and 565 pairs
38 basis elements and 564 pairs
38 basis elements and 563 pairs
38 basis elements and 562 pairs
38 basis elements and 561 pairs
38 basis elements and 560 pairs
38 basis elements and 559 pairs
39 basis elements and 596 pairs
40 basis elements and 634 pairs
40 basis elements and 633 pairs
40 basis elements and 632 pairs
40 basis elements and 631 pairs
40 basis elements and 630 pairs
40 basis elements and 629 pairs
40 basis elements and 628 pairs
41 basis elements and 667 pairs
41 basis elements and 666 pairs
```

```
41 basis elements and 665 pairs
41 basis elements and 664 pairs
41 basis elements and 663 pairs
41 basis elements and 662 pairs
41 basis elements and 661 pairs
41 basis elements and 660 pairs
41 basis elements and 659 pairs
41 basis elements and 658 pairs
41 basis elements and 657 pairs
42 basis elements and 697 pairs
42 basis elements and 696 pairs
42 basis elements and 695 pairs
43 basis elements and 736 pairs
43 basis elements and 735 pairs
43 basis elements and 734 pairs
44 basis elements and 776 pairs
44 basis elements and 775 pairs
44 basis elements and 774 pairs
44 basis elements and 773 pairs
44 basis elements and 772 pairs
44 basis elements and 771 pairs
44 basis elements and 770 pairs
44 basis elements and 769 pairs
44 basis elements and 768 pairs
44 basis elements and 767 pairs
44 basis elements and 766 pairs
44 basis elements and 765 pairs
44 basis elements and 764 pairs
44 basis elements and 763 pairs
44 basis elements and 762 pairs
44 basis elements and 761 pairs
44 basis elements and 760 pairs
44 basis elements and 759 pairs
44 basis elements and 758 pairs
44 basis elements and 757 pairs
45 basis elements and 800 pairs
45 basis elements and 799 pairs
45 basis elements and 798 pairs
45 basis elements and 797 pairs
45 basis elements and 796 pairs
45 basis elements and 795 pairs
45 basis elements and 794 pairs
45 basis elements and 793 pairs
45 basis elements and 792 pairs
45 basis elements and 791 pairs
45 basis elements and 790 pairs
45 basis elements and 789 pairs
45 basis elements and 788 pairs
45 basis elements and 787 pairs
45 basis elements and 786 pairs
45 basis elements and 785 pairs
45 basis elements and 784 pairs
45 basis elements and 783 pairs
45 basis elements and 782 pairs
45 basis elements and 781 pairs
```

```
CPU time (user): 0.0120009999999
- : bool = true
// grobner.p018
2 basis elements and 1 pairs
CPU time (user): 0.
- : bool = true
// grobner.p019
CPU time (user): 1.740109
- : fol formula = <<true>>
// grobner.p020
4 basis elements and 6 pairs
4 basis elements and 5 pairs
4 basis elements and 4 pairs
CPU time (user): 0.
- : bool = true
// grobner.p021
CPU time (user): 0.044002
- : fol formula = <<true>>
// grobner.p022
3 basis elements and 3 pairs
3 basis elements and 2 pairs
CPU time (user): 0.00400100000002
- : bool = true
// grobner.p023
4 basis elements and 6 pairs
4 basis elements and 5 pairs
4 basis elements and 4 pairs
4 basis elements and 3 pairs
5 basis elements and 6 pairs
6 basis elements and 10 pairs
7 basis elements and 15 pairs
7 basis elements and 14 pairs
8 basis elements and 20 pairs
8 basis elements and 19 pairs
CPU time (user): 0.
- : bool = true
// grobner.p024
4 basis elements and 6 pairs
4 basis elements and 5 pairs
4 basis elements and 4 pairs
4 basis elements and 3 pairs
5 basis elements and 6 pairs
6 basis elements and 10 pairs
7 basis elements and 15 pairs
7 basis elements and 14 pairs
8 basis elements and 20 pairs
8 basis elements and 19 pairs
CPU time (user): 0.
- : bool = true
// grobner.p025
4 basis elements and 6 pairs
4 basis elements and 5 pairs
4 basis elements and 4 pairs
4 basis elements and 3 pairs
5 basis elements and 6 pairs
```

```
6 basis elements and 10 pairs
7 basis elements and 15 pairs
7 basis elements and 14 pairs
8 basis elements and 20 pairs
8 basis elements and 19 pairs
CPU time (user): 0.00400000000002
- : bool = true
// grobner.p026
4 basis elements and 6 pairs
4 basis elements and 5 pairs
4 basis elements and 4 pairs
4 basis elements and 3 pairs
5 basis elements and 6 pairs
6 basis elements and 10 pairs
7 basis elements and 15 pairs
7 basis elements and 14 pairs
8 basis elements and 20 pairs
8 basis elements and 19 pairs
CPU time (user): 0.
- : bool = true
// grobner.p027
3 basis elements and 3 pairs
4 basis elements and 5 pairs
5 basis elements and 8 pairs
6 basis elements and 12 pairs
7 basis elements and 17 pairs
7 basis elements and 16 pairs
CPU time (user): 0.00400000000002
- : bool = true
- : dummy_interactive = END_INTERACTIVE
#
```

geom.ml

```
#use geom.ml";;
```

```
# #use "geom.ml";;
val coordinations : (string * fol formula) list =
  [("collinear",
    <<(1_x - 2_x) * (2_y - 3_y) = (1_y - 2_y) * (2_x - 3_x)>>);
   ("parallel",
    <<(1_x - 2_x) * (3_y - 4_y) = (1_y - 2_y) * (3_x - 4_x)>>);
   ("perpendicular",
    <<(1_x - 2_x) * (3_x - 4_x) + (1_y - 2_y) * (3_y - 4_y) = 0>>);
   ("lengths_eq",
    <<(1_x - 2_x)^2 + (1_y - 2_y)^2 = (3_x - 4_x)^2 + (3_y - 4_y)^2>>);
   ("is_midpoint", <<2 * 1_x = 2_x + 3_x /\ 2 * 1_y = 2_y + 3_y>>);
   ("is_intersection",
    <<(1_x - 2_x) * (2_y - 3_y) = (1_y - 2_y) * (2_x - 3_x) /\
      (1_x - 4_x) * (4_y - 5_y) = (1_y - 4_y) * (4_x - 5_x)>>);
   ("=", <<1_x = 2_x /\ 1_y = 2_y>>)]
File "/home/student/Harrison/OCaml/geom.ml", line 35, characters 11-49:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
Fn (_, _)
val coordinate : fol formula -> fol formula = <fun>
- : dummy_interactive = START_INTERACTIVE
// geom.p001
- : fol formula =
<<(a_x - b_x) * (b_y - c_y) = (a_y - b_y) * (b_x - c_x) ==>
  (b_x - a_x) * (a_y - c_y) = (b_y - a_y) * (a_x - c_x)>>
- : dummy_interactive = END_INTERACTIVE
val invariant : term * term -> string * fol formula -> fol formula =
  <fun>
val invariant_under_translation : string * fol formula -> fol formula =
  <fun>
- : dummy_interactive = START_INTERACTIVE
// geom.p002
2 basis elements and 1 pairs
2 basis elements and 1 pairs
2 basis elements and 1 pairs
2 basis elements and 1 pairs
2 basis elements and 1 pairs
2 basis elements and 1 pairs
2 basis elements and 1 pairs
2 basis elements and 1 pairs
3 basis elements and 3 pairs
3 basis elements and 2 pairs
3 basis elements and 3 pairs
3 basis elements and 2 pairs
4 basis elements and 4 pairs
3 basis elements and 3 pairs
3 basis elements and 2 pairs
```

```
3 basis elements and 3 pairs
3 basis elements and 2 pairs
4 basis elements and 4 pairs
3 basis elements and 3 pairs
3 basis elements and 2 pairs
3 basis elements and 3 pairs
3 basis elements and 2 pairs
4 basis elements and 4 pairs
3 basis elements and 3 pairs
3 basis elements and 2 pairs
3 basis elements and 3 pairs
3 basis elements and 2 pairs
4 basis elements and 4 pairs
3 basis elements and 3 pairs
3 basis elements and 2 pairs
3 basis elements and 3 pairs
3 basis elements and 2 pairs
4 basis elements and 4 pairs
3 basis elements and 3 pairs
3 basis elements and 2 pairs
3 basis elements and 3 pairs
3 basis elements and 2 pairs
4 basis elements and 4 pairs
- : bool = true
- : dummy_interactive = END_INTERACTIVE
val invariant_under_rotation : string * fol formula -> fol formula =
  <fun>
- : dummy_interactive = START_INTERACTIVE
// geom.p003
3 basis elements and 3 pairs
3 basis elements and 2 pairs
3 basis elements and 3 pairs
4 basis elements and 5 pairs
5 basis elements and 8 pairs
5 basis elements and 7 pairs
5 basis elements and 6 pairs
5 basis elements and 5 pairs
3 basis elements and 3 pairs
3 basis elements and 2 pairs
3 basis elements and 3 pairs
4 basis elements and 5 pairs
5 basis elements and 8 pairs
5 basis elements and 7 pairs
5 basis elements and 6 pairs
5 basis elements and 5 pairs
3 basis elements and 3 pairs
3 basis elements and 2 pairs
3 basis elements and 3 pairs
4 basis elements and 5 pairs
5 basis elements and 8 pairs
5 basis elements and 7 pairs
5 basis elements and 6 pairs
5 basis elements and 5 pairs
3 basis elements and 3 pairs
3 basis elements and 2 pairs
```

```
3 basis elements and 3 pairs
4 basis elements and 5 pairs
5 basis elements and 8 pairs
5 basis elements and 7 pairs
5 basis elements and 6 pairs
5 basis elements and 5 pairs
4 basis elements and 6 pairs
4 basis elements and 5 pairs
4 basis elements and 4 pairs
5 basis elements and 7 pairs
5 basis elements and 6 pairs
4 basis elements and 6 pairs
4 basis elements and 5 pairs
4 basis elements and 4 pairs
5 basis elements and 7 pairs
5 basis elements and 6 pairs
4 basis elements and 6 pairs
5 basis elements and 9 pairs
5 basis elements and 8 pairs
6 basis elements and 12 pairs
7 basis elements and 17 pairs
8 basis elements and 23 pairs
9 basis elements and 30 pairs
9 basis elements and 29 pairs
9 basis elements and 28 pairs
9 basis elements and 27 pairs
4 basis elements and 6 pairs
5 basis elements and 9 pairs
5 basis elements and 8 pairs
6 basis elements and 12 pairs
7 basis elements and 17 pairs
8 basis elements and 23 pairs
9 basis elements and 30 pairs
9 basis elements and 29 pairs
9 basis elements and 28 pairs
10 basis elements and 36 pairs
10 basis elements and 35 pairs
10 basis elements and 34 pairs
10 basis elements and 33 pairs
10 basis elements and 32 pairs
10 basis elements and 31 pairs
10 basis elements and 30 pairs
10 basis elements and 29 pairs
10 basis elements and 28 pairs
10 basis elements and 27 pairs
10 basis elements and 26 pairs
11 basis elements and 35 pairs
4 basis elements and 6 pairs
4 basis elements and 5 pairs
4 basis elements and 4 pairs
4 basis elements and 6 pairs
4 basis elements and 5 pairs
4 basis elements and 4 pairs
5 basis elements and 7 pairs
5 basis elements and 6 pairs
```

```
4 basis elements and 6 pairs
4 basis elements and 5 pairs
5 basis elements and 8 pairs
6 basis elements and 12 pairs
7 basis elements and 17 pairs
7 basis elements and 16 pairs
7 basis elements and 15 pairs
7 basis elements and 14 pairs
7 basis elements and 13 pairs
7 basis elements and 12 pairs
4 basis elements and 6 pairs
4 basis elements and 5 pairs
5 basis elements and 8 pairs
6 basis elements and 12 pairs
7 basis elements and 17 pairs
8 basis elements and 23 pairs
8 basis elements and 22 pairs
8 basis elements and 21 pairs
8 basis elements and 20 pairs
8 basis elements and 19 pairs
8 basis elements and 18 pairs
8 basis elements and 17 pairs
8 basis elements and 16 pairs
8 basis elements and 15 pairs
8 basis elements and 14 pairs
9 basis elements and 21 pairs
9 basis elements and 20 pairs
10 basis elements and 28 pairs
10 basis elements and 27 pairs
10 basis elements and 26 pairs
10 basis elements and 25 pairs
4 basis elements and 6 pairs
4 basis elements and 5 pairs
4 basis elements and 4 pairs
5 basis elements and 7 pairs
5 basis elements and 6 pairs
4 basis elements and 6 pairs
4 basis elements and 5 pairs
4 basis elements and 4 pairs
5 basis elements and 7 pairs
5 basis elements and 6 pairs
4 basis elements and 6 pairs
4 basis elements and 5 pairs
5 basis elements and 8 pairs
6 basis elements and 12 pairs
7 basis elements and 17 pairs
8 basis elements and 23 pairs
9 basis elements and 30 pairs
9 basis elements and 29 pairs
9 basis elements and 28 pairs
9 basis elements and 27 pairs
4 basis elements and 6 pairs
4 basis elements and 5 pairs
5 basis elements and 8 pairs
6 basis elements and 12 pairs
```

```
7 basis elements and 17 pairs
- : bool = true
- : dummy_interactive = END_INTERACTIVE
- : dummy_interactive = START_INTERACTIVE
// geom.p004
- : fol formula = <<true>>
- : dummy_interactive = END_INTERACTIVE
File "/home/student/Harrison/OCaml/geom.ml", line 87, characters 6-15:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
[]
val originate : fol formula -> fol formula = <fun>
val invariant_under_scaling : string * fol formula -> fol formula =
  <fun>
val invariant_under_shearing : string * fol formula -> fol formula =
  <fun>
- : dummy_interactive = START_INTERACTIVE
// geom.p005
3 basis elements and 3 pairs
3 basis elements and 2 pairs
3 basis elements and 3 pairs
4 basis elements and 5 pairs
5 basis elements and 8 pairs
5 basis elements and 7 pairs
5 basis elements and 6 pairs
6 basis elements and 10 pairs
7 basis elements and 15 pairs
7 basis elements and 14 pairs
7 basis elements and 13 pairs
7 basis elements and 12 pairs
7 basis elements and 11 pairs
7 basis elements and 10 pairs
7 basis elements and 9 pairs
7 basis elements and 8 pairs
8 basis elements and 14 pairs
3 basis elements and 3 pairs
3 basis elements and 2 pairs
3 basis elements and 3 pairs
4 basis elements and 5 pairs
5 basis elements and 8 pairs
5 basis elements and 7 pairs
5 basis elements and 6 pairs
6 basis elements and 10 pairs
7 basis elements and 15 pairs
7 basis elements and 14 pairs
7 basis elements and 13 pairs
7 basis elements and 12 pairs
7 basis elements and 11 pairs
7 basis elements and 10 pairs
7 basis elements and 9 pairs
7 basis elements and 8 pairs
8 basis elements and 14 pairs
3 basis elements and 3 pairs
3 basis elements and 2 pairs
3 basis elements and 3 pairs
```

```
4 basis elements and 5 pairs
5 basis elements and 8 pairs
5 basis elements and 7 pairs
5 basis elements and 6 pairs
6 basis elements and 10 pairs
7 basis elements and 15 pairs
7 basis elements and 14 pairs
7 basis elements and 13 pairs
7 basis elements and 12 pairs
7 basis elements and 11 pairs
7 basis elements and 10 pairs
7 basis elements and 9 pairs
7 basis elements and 8 pairs
8 basis elements and 14 pairs
3 basis elements and 3 pairs
3 basis elements and 2 pairs
3 basis elements and 3 pairs
4 basis elements and 5 pairs
5 basis elements and 8 pairs
5 basis elements and 7 pairs
5 basis elements and 6 pairs
6 basis elements and 10 pairs
7 basis elements and 15 pairs
7 basis elements and 14 pairs
7 basis elements and 13 pairs
7 basis elements and 12 pairs
7 basis elements and 11 pairs
7 basis elements and 10 pairs
7 basis elements and 9 pairs
7 basis elements and 8 pairs
8 basis elements and 14 pairs
4 basis elements and 6 pairs
4 basis elements and 5 pairs
4 basis elements and 4 pairs
4 basis elements and 6 pairs
4 basis elements and 5 pairs
4 basis elements and 4 pairs
5 basis elements and 7 pairs
5 basis elements and 6 pairs
4 basis elements and 6 pairs
4 basis elements and 5 pairs
5 basis elements and 8 pairs
6 basis elements and 12 pairs
7 basis elements and 17 pairs
7 basis elements and 16 pairs
8 basis elements and 22 pairs
8 basis elements and 21 pairs
8 basis elements and 20 pairs
9 basis elements and 27 pairs
4 basis elements and 6 pairs
4 basis elements and 5 pairs
5 basis elements and 8 pairs
6 basis elements and 12 pairs
7 basis elements and 17 pairs
8 basis elements and 23 pairs
```

```
9 basis elements and 30 pairs
9 basis elements and 29 pairs
9 basis elements and 28 pairs
10 basis elements and 36 pairs
10 basis elements and 35 pairs
10 basis elements and 34 pairs
10 basis elements and 33 pairs
10 basis elements and 32 pairs
11 basis elements and 41 pairs
11 basis elements and 40 pairs
11 basis elements and 39 pairs
11 basis elements and 38 pairs
11 basis elements and 37 pairs
11 basis elements and 36 pairs
12 basis elements and 46 pairs
4 basis elements and 6 pairs
4 basis elements and 5 pairs
4 basis elements and 4 pairs
4 basis elements and 6 pairs
4 basis elements and 5 pairs
4 basis elements and 4 pairs
5 basis elements and 7 pairs
5 basis elements and 6 pairs
4 basis elements and 6 pairs
4 basis elements and 5 pairs
5 basis elements and 8 pairs
6 basis elements and 12 pairs
7 basis elements and 17 pairs
7 basis elements and 16 pairs
7 basis elements and 15 pairs
7 basis elements and 14 pairs
7 basis elements and 13 pairs
8 basis elements and 19 pairs
9 basis elements and 26 pairs
9 basis elements and 25 pairs
9 basis elements and 24 pairs
9 basis elements and 23 pairs
9 basis elements and 22 pairs
9 basis elements and 21 pairs
9 basis elements and 20 pairs
9 basis elements and 19 pairs
9 basis elements and 18 pairs
9 basis elements and 17 pairs
10 basis elements and 25 pairs
10 basis elements and 24 pairs
10 basis elements and 23 pairs
10 basis elements and 22 pairs
10 basis elements and 21 pairs
10 basis elements and 20 pairs
10 basis elements and 19 pairs
11 basis elements and 28 pairs
4 basis elements and 6 pairs
4 basis elements and 5 pairs
5 basis elements and 8 pairs
6 basis elements and 12 pairs
```

```
7 basis elements and 17 pairs
8 basis elements and 23 pairs
8 basis elements and 22 pairs
8 basis elements and 21 pairs
8 basis elements and 20 pairs
9 basis elements and 27 pairs
10 basis elements and 35 pairs
10 basis elements and 34 pairs
10 basis elements and 33 pairs
10 basis elements and 32 pairs
11 basis elements and 41 pairs
11 basis elements and 40 pairs
11 basis elements and 39 pairs
12 basis elements and 49 pairs
12 basis elements and 48 pairs
12 basis elements and 47 pairs
13 basis elements and 58 pairs
14 basis elements and 70 pairs
14 basis elements and 69 pairs
14 basis elements and 68 pairs
14 basis elements and 67 pairs
14 basis elements and 66 pairs
14 basis elements and 65 pairs
14 basis elements and 64 pairs
14 basis elements and 63 pairs
14 basis elements and 62 pairs
14 basis elements and 61 pairs
14 basis elements and 60 pairs
14 basis elements and 59 pairs
14 basis elements and 58 pairs
14 basis elements and 57 pairs
15 basis elements and 70 pairs
15 basis elements and 69 pairs
15 basis elements and 68 pairs
15 basis elements and 67 pairs
15 basis elements and 66 pairs
15 basis elements and 65 pairs
15 basis elements and 64 pairs
15 basis elements and 63 pairs
15 basis elements and 62 pairs
16 basis elements and 76 pairs
16 basis elements and 75 pairs
16 basis elements and 74 pairs
16 basis elements and 73 pairs
16 basis elements and 72 pairs
16 basis elements and 71 pairs
16 basis elements and 70 pairs
16 basis elements and 69 pairs
16 basis elements and 68 pairs
16 basis elements and 67 pairs
17 basis elements and 82 pairs
17 basis elements and 81 pairs
17 basis elements and 80 pairs
17 basis elements and 79 pairs
17 basis elements and 78 pairs
```

```
17 basis elements and 77 pairs
17 basis elements and 76 pairs
17 basis elements and 75 pairs
17 basis elements and 74 pairs
17 basis elements and 73 pairs
17 basis elements and 72 pairs
18 basis elements and 88 pairs
18 basis elements and 87 pairs
18 basis elements and 86 pairs
18 basis elements and 85 pairs
18 basis elements and 84 pairs
18 basis elements and 83 pairs
18 basis elements and 82 pairs
18 basis elements and 81 pairs
18 basis elements and 80 pairs
18 basis elements and 79 pairs
18 basis elements and 78 pairs
18 basis elements and 77 pairs
19 basis elements and 94 pairs
4 basis elements and 6 pairs
4 basis elements and 5 pairs
4 basis elements and 4 pairs
4 basis elements and 6 pairs
4 basis elements and 5 pairs
4 basis elements and 4 pairs
5 basis elements and 7 pairs
5 basis elements and 6 pairs
4 basis elements and 6 pairs
4 basis elements and 5 pairs
5 basis elements and 8 pairs
6 basis elements and 12 pairs
6 basis elements and 11 pairs
7 basis elements and 16 pairs
7 basis elements and 15 pairs
7 basis elements and 14 pairs
7 basis elements and 13 pairs
7 basis elements and 12 pairs
4 basis elements and 6 pairs
4 basis elements and 5 pairs
5 basis elements and 8 pairs
6 basis elements and 12 pairs
7 basis elements and 17 pairs
8 basis elements and 23 pairs
8 basis elements and 22 pairs
8 basis elements and 21 pairs
8 basis elements and 20 pairs
8 basis elements and 19 pairs
9 basis elements and 26 pairs
9 basis elements and 25 pairs
9 basis elements and 24 pairs
9 basis elements and 23 pairs
9 basis elements and 22 pairs
9 basis elements and 21 pairs
9 basis elements and 20 pairs
9 basis elements and 19 pairs
```

```
9 basis elements and 18 pairs
9 basis elements and 17 pairs
9 basis elements and 16 pairs
- : bool = true
// geom.p006 - TODO: was all of result copied?
3 basis elements and 3 pairs
3 basis elements and 2 pairs
4 basis elements and 4 pairs
3 basis elements and 3 pairs
4 basis elements and 5 pairs
5 basis elements and 8 pairs
5 basis elements and 7 pairs
5 basis elements and 6 pairs
5 basis elements and 5 pairs
3 basis elements and 3 pairs
3 basis elements and 2 pairs
3 basis elements and 3 pairs
3 basis elements and 2 pairs
4 basis elements and 4 pairs
3 basis elements and 3 pairs
3 basis elements and 2 pairs
3 basis elements and 3 pairs
3 basis elements and 2 pairs
4 basis elements and 4 pairs
3 basis elements and 3 pairs
3 basis elements and 2 pairs
4 basis elements and 4 pairs
3 basis elements and 3 pairs
4 basis elements and 5 pairs
5 basis elements and 8 pairs
5 basis elements and 7 pairs
5 basis elements and 6 pairs
5 basis elements and 5 pairs
2 basis elements and 1 pairs
3 basis elements and 2 pairs
3 basis elements and 1 pairs
3 basis elements and 0 pairs
2 basis elements and 1 pairs
3 basis elements and 2 pairs
3 basis elements and 1 pairs
3 basis elements and 0 pairs
2 basis elements and 1 pairs
2 basis elements and 1 pairs
2 basis elements and 1 pairs
2 basis elements and 1 pairs
- : (string * fol formula) list * (string * fol formula) list =
([("collinear",
   <<(1_x - 2_x) * (2_y - 3_y) = (1_y - 2_y) * (2_x - 3_x)>>);
  ("parallel",
   <<(1_x - 2_x) * (3_y - 4_y) = (1_y - 2_y) * (3_x - 4_x)>>);
  ("is_midpoint", <<2 * 1_x = 2_x + 3_x /\ 2 * 1_y = 2_y + 3_y>>);
  ("is_intersection",
   <<(1_x - 2_x) * (2_y - 3_y) = (1_y - 2_y) * (2_x - 3_x) /\
     (1_x - 4_x) * (4_y - 5_y) = (1_y - 4_y) * (4_x - 5_x)>>);
  ("=", <<1_x = 2_x /\ 1_y = 2_y>>)],
 [("perpendicular",
   <<(1_x - 2_x) * (3_x - 4_x) + (1_y - 2_y) * (3_y - 4_y) = 0>>);
  ("lengths_eq",
   <<(1_x - 2_x)^2 + (1_y - 2_y)^2 = (3_x - 4_x)^2 + (3_y - 4_y)^2>>)])
- : dummy_interactive = END_INTERACTIVE
- : dummy_interactive = START_INTERACTIVE
// geom.p007
4 basis elements and 6 pairs
4 basis elements and 5 pairs
5 basis elements and 8 pairs
6 basis elements and 12 pairs
7 basis elements and 17 pairs
8 basis elements and 23 pairs
8 basis elements and 22 pairs
8 basis elements and 21 pairs
8 basis elements and 20 pairs
8 basis elements and 19 pairs
8 basis elements and 18 pairs
8 basis elements and 17 pairs
8 basis elements and 16 pairs
8 basis elements and 15 pairs
8 basis elements and 14 pairs
8 basis elements and 13 pairs
8 basis elements and 12 pairs
8 basis elements and 11 pairs
8 basis elements and 10 pairs
8 basis elements and 9 pairs
8 basis elements and 8 pairs
- : bool = true
// geom.p008
5 basis elements and 10 pairs
6 basis elements and 14 pairs
7 basis elements and 19 pairs
7 basis elements and 18 pairs
8 basis elements and 24 pairs
8 basis elements and 23 pairs
9 basis elements and 30 pairs
10 basis elements and 38 pairs
10 basis elements and 37 pairs
11 basis elements and 46 pairs
12 basis elements and 56 pairs
12 basis elements and 55 pairs
12 basis elements and 54 pairs
12 basis elements and 53 pairs
12 basis elements and 52 pairs
13 basis elements and 63 pairs
13 basis elements and 62 pairs
13 basis elements and 61 pairs
13 basis elements and 60 pairs
13 basis elements and 59 pairs
13 basis elements and 58 pairs
13 basis elements and 57 pairs
13 basis elements and 56 pairs
13 basis elements and 55 pairs
13 basis elements and 54 pairs
```

```
13 basis elements and 53 pairs
13 basis elements and 52 pairs
13 basis elements and 51 pairs
13 basis elements and 50 pairs
13 basis elements and 49 pairs
13 basis elements and 48 pairs
13 basis elements and 47 pairs
13 basis elements and 46 pairs
13 basis elements and 45 pairs
13 basis elements and 44 pairs
13 basis elements and 43 pairs
13 basis elements and 42 pairs
13 basis elements and 41 pairs
13 basis elements and 40 pairs
13 basis elements and 39 pairs
13 basis elements and 38 pairs
13 basis elements and 37 pairs
13 basis elements and 36 pairs
13 basis elements and 35 pairs
13 basis elements and 34 pairs
13 basis elements and 33 pairs
13 basis elements and 32 pairs
13 basis elements and 31 pairs
13 basis elements and 30 pairs
13 basis elements and 29 pairs
13 basis elements and 28 pairs
13 basis elements and 27 pairs
13 basis elements and 26 pairs
13 basis elements and 25 pairs
13 basis elements and 24 pairs
13 basis elements and 23 pairs
13 basis elements and 22 pairs
13 basis elements and 21 pairs
13 basis elements and 20 pairs
13 basis elements and 19 pairs
13 basis elements and 18 pairs
13 basis elements and 17 pairs
13 basis elements and 16 pairs
13 basis elements and 15 pairs
13 basis elements and 14 pairs
13 basis elements and 13 pairs
13 basis elements and 12 pairs
13 basis elements and 11 pairs
13 basis elements and 10 pairs
14 basis elements and 22 pairs
15 basis elements and 35 pairs
15 basis elements and 34 pairs
15 basis elements and 33 pairs
15 basis elements and 32 pairs
15 basis elements and 31 pairs
15 basis elements and 30 pairs
15 basis elements and 29 pairs
15 basis elements and 28 pairs
15 basis elements and 27 pairs
15 basis elements and 26 pairs
```

```
15 basis elements and 25 pairs
15 basis elements and 24 pairs
15 basis elements and 23 pairs
15 basis elements and 22 pairs
15 basis elements and 21 pairs
15 basis elements and 20 pairs
15 basis elements and 19 pairs
15 basis elements and 18 pairs
15 basis elements and 17 pairs
15 basis elements and 16 pairs
15 basis elements and 15 pairs
15 basis elements and 14 pairs
15 basis elements and 13 pairs
15 basis elements and 12 pairs
15 basis elements and 11 pairs
15 basis elements and 10 pairs
15 basis elements and 9 pairs
15 basis elements and 8 pairs
15 basis elements and 7 pairs
15 basis elements and 6 pairs
15 basis elements and 5 pairs
15 basis elements and 4 pairs
15 basis elements and 3 pairs
15 basis elements and 2 pairs
15 basis elements and 1 pairs
15 basis elements and 0 pairs
- : bool = false
// geom.p009
6 basis elements and 15 pairs
7 basis elements and 20 pairs
8 basis elements and 26 pairs
8 basis elements and 25 pairs
9 basis elements and 32 pairs
10 basis elements and 40 pairs
10 basis elements and 39 pairs
11 basis elements and 48 pairs
12 basis elements and 58 pairs
13 basis elements and 69 pairs
13 basis elements and 68 pairs
14 basis elements and 80 pairs
15 basis elements and 93 pairs
16 basis elements and 107 pairs
17 basis elements and 122 pairs
18 basis elements and 138 pairs
18 basis elements and 137 pairs
18 basis elements and 136 pairs
18 basis elements and 135 pairs
18 basis elements and 134 pairs
19 basis elements and 151 pairs
20 basis elements and 169 pairs
20 basis elements and 168 pairs
20 basis elements and 167 pairs
20 basis elements and 166 pairs
20 basis elements and 165 pairs
20 basis elements and 164 pairs
```

```
20 basis elements and 163 pairs
20 basis elements and 162 pairs
20 basis elements and 161 pairs
20 basis elements and 160 pairs
21 basis elements and 179 pairs
21 basis elements and 178 pairs
21 basis elements and 177 pairs
22 basis elements and 197 pairs
22 basis elements and 196 pairs
22 basis elements and 195 pairs
22 basis elements and 194 pairs
22 basis elements and 193 pairs
22 basis elements and 192 pairs
22 basis elements and 191 pairs
22 basis elements and 190 pairs
22 basis elements and 189 pairs
22 basis elements and 188 pairs
22 basis elements and 187 pairs
22 basis elements and 186 pairs
22 basis elements and 185 pairs
22 basis elements and 184 pairs
22 basis elements and 183 pairs
22 basis elements and 182 pairs
22 basis elements and 181 pairs
22 basis elements and 180 pairs
22 basis elements and 179 pairs
22 basis elements and 178 pairs
22 basis elements and 177 pairs
22 basis elements and 176 pairs
22 basis elements and 175 pairs
22 basis elements and 174 pairs
23 basis elements and 195 pairs
23 basis elements and 194 pairs
23 basis elements and 193 pairs
24 basis elements and 215 pairs
24 basis elements and 214 pairs
24 basis elements and 213 pairs
24 basis elements and 212 pairs
25 basis elements and 235 pairs
25 basis elements and 234 pairs
25 basis elements and 233 pairs
25 basis elements and 232 pairs
25 basis elements and 231 pairs
25 basis elements and 230 pairs
25 basis elements and 229 pairs
25 basis elements and 228 pairs
25 basis elements and 227 pairs
25 basis elements and 226 pairs
25 basis elements and 225 pairs
25 basis elements and 224 pairs
25 basis elements and 223 pairs
25 basis elements and 222 pairs
25 basis elements and 221 pairs
25 basis elements and 220 pairs
25 basis elements and 219 pairs
```

```
25 basis elements and 218 pairs
25 basis elements and 217 pairs
25 basis elements and 216 pairs
25 basis elements and 215 pairs
25 basis elements and 214 pairs
25 basis elements and 213 pairs
25 basis elements and 212 pairs
25 basis elements and 211 pairs
26 basis elements and 235 pairs
26 basis elements and 234 pairs
26 basis elements and 233 pairs
26 basis elements and 232 pairs
26 basis elements and 231 pairs
26 basis elements and 230 pairs
26 basis elements and 229 pairs
26 basis elements and 228 pairs
26 basis elements and 227 pairs
26 basis elements and 226 pairs
26 basis elements and 225 pairs
26 basis elements and 224 pairs
26 basis elements and 223 pairs
26 basis elements and 222 pairs
- : bool = true
- : dummy_interactive = END_INTERACTIVE
File "/home/student/Harrison/OCaml/geom.ml", line 136, characters 2-522:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
Var _::_
val pprove :
  string list ->
  term list -> term -> fol formula list -> fol formula list = <fun>
val triangulate : string list -> term list -> term list -> term list =
  <fun>
val wu : fol formula -> string list -> string list -> fol formula list =
  <fun>
- : dummy_interactive = START_INTERACTIVE
val simson : fol formula =
  <<lengths_eq(o,a,o,b) /\
    lengths_eq(o,a,o,c) /\
    lengths_eq(o,a,o,d) /\
    collinear(e,b,c) /\
    collinear(f,a,c) /\
    collinear(g,a,b) /\
    perpendicular(b,c,d,e) /\
    perpendicular(a,c,d,f) /\ perpendicular(a,b,d,g) ==>
    collinear(e,f,g)>>
val vars : string list =
  ["g_y"; "g_x"; "f_y"; "f_x"; "e_y"; "e_x"; "d_y"; "d_x"; "c_y";
   "c_x"; "b_y"; "b_x"; "o_x"]
val zeros : string list = ["a_x"; "a_y"; "o_y"]
// geom.p010
- : fol formula list =
[<<~(((0 + b_x * (0 + b_x * 1)) + b_y * (0 + b_y * 1)) + c_x *
     ((0 + b_x * -2) + c_x * 1)) +
    c_y * ((0 + b_y * -2) + c_y * 1) = 0>>;
```

```
 <<~(0 + b_x * (0 + b_x * 1)) + b_y * (0 + b_y * 1) = 0>>;
 <<~(0 + b_x * -1) + c_x * 1 = 0>>;
 <<~(0 + c_x * (0 + c_x * 1)) + c_y * (0 + c_y * 1) = 0>>;
 <<~0 + b_x * 1 = 0>>; <<~0 + c_x * 1 = 0>>; <<~-1 = 0>>]
// geom.p011
- : fol formula list =
[<<~(((0 + a_y * (0 + a_y * 1)) + a_x * (0 + a_x * 1)) + b_x *
     ((0 + a_x * -2) + b_x * 1)) +
    b_y * ((0 + a_y * -2) + b_y * 1) = 0>>;
 <<~(((0 + a_y * (0 + a_y * 1)) + a_x * (0 + a_x * 1)) + c_x *
     ((0 + a_x * -2) + c_x * 1)) +
    c_y * ((0 + a_y * -2) + c_y * 1) = 0>>;
 <<~(((0 + b_x * (0 + b_x * 1)) + b_y * (0 + b_y * 1)) + c_x *
     ((0 + b_x * -2) + c_x * 1)) +
    c_y * ((0 + b_y * -2) + c_y * 1) = 0>>;
 <<~(0 + a_x * -1) + b_x * 1 = 0>>; <<~(0 + a_x * -1) + c_x * 1 = 0>>;
 <<~(0 + b_x * -1) + c_x * 1 = 0>>; <<~-1 = 0>>]
val pappus : fol formula =
  <<collinear(a1,b2,d) /\
    collinear(a2,b1,d) /\
    collinear(a2,b3,e) /\
    collinear(a3,b2,e) /\ collinear(a1,b3,f) /\ collinear(a3,b1,f) ==>
    collinear(d,e,f)>>
val vars : string list =
  ["f_y"; "f_x"; "e_y"; "e_x"; "d_y"; "d_x"; "b3_y"; "b2_y"; "b1_y";
   "a3_x"; "a2_x"; "a1_x"]
val zeros : string list =
  ["a1_y"; "a2_y"; "a3_y"; "b1_x"; "b2_x"; "b3_x"]
// geom.p012
- : fol formula list =
[<<~(0 + b1_y * (0 + a1_x * 1)) + b2_y * (0 + a2_x * -1) = 0>>;
 <<~(0 + b1_y * (0 + a1_x * 1)) + b3_y * (0 + a3_x * -1) = 0>>;
 <<~(0 + b2_y * (0 + a2_x * 1)) + b3_y * (0 + a3_x * -1) = 0>>;
 <<~0 + a1_x * -1 = 0>>; <<~0 + a2_x * -1 = 0>>]
- : dummy_interactive = END_INTERACTIVE
#
```

interpolation.ml

```
#use interpolation.ml";;
```

```
# #use "interpolation.ml";;
val pinterpolate : 'a formula -> 'a formula -> 'a formula = <fun>
File "/home/student/Harrison/OCaml/interpolation.ml", line 28, characters 26-47:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
(Exists (_, _)|Forall (_, _)|Iff (_, _)|Imp (_, _)|Or (_, _)|Not _|Atom _|
True|False)
val urinterpolate : fol formula -> fol formula -> fol formula = <fun>
- : dummy_interactive = START_INTERACTIVE
// interpolation.p001
val p : fol formula =
  <<forall x y. R(x,f(x)) /\ (S(x,y) <=> R(x,y) \/ R(y,x))>>
// interpolation.p002
val q : fol formula =
  <<forall x y z. (S(x,y) /\ S(y,z) ==> T(x,z)) /\ ~T(0,0)>>
// interpolation.p003
0 ground instances tried; 0 items in list
0 ground instances tried; 0 items in list
1 ground instances tried; 5 items in list
1 ground instances tried; 5 items in list
2 ground instances tried; 6 items in list
3 ground instances tried; 10 items in list
val c : fol formula =
  <<S(0,f(0)) /\ S(f(0),0) \/ S(0,f(0)) /\ S(f(0),0)>>
// interpolation.p004
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
- : int list = [2; 2]
// interpolation.p005
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
- : int list = [3]
- : dummy_interactive = END_INTERACTIVE
val toptermt : (string * int) list -> term -> term list = <fun>
val topterms : (string * int) list -> fol formula -> term list = <fun>
File "/home/student/Harrison/OCaml/interpolation.ml", line 67, characters 4-287:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
Var _::_
val uinterpolate : fol formula -> fol formula -> fol formula = <fun>
- : dummy_interactive = START_INTERACTIVE
// interpolation.p006
0 ground instances tried; 0 items in list
```

```
0 ground instances tried; 0 items in list
1 ground instances tried; 5 items in list
1 ground instances tried; 5 items in list
2 ground instances tried; 6 items in list
3 ground instances tried; 10 items in list
val c : fol formula =
  <<forall v_2.
      exists v_1. S(v_2,v_1) /\ S(v_1,v_2) \/ S(v_2,v_1) /\ S(v_1,v_2)>>
// interpolation.p007
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
- : int list = [4]
// interpolation.p008
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
- : int list = [3]
- : dummy_interactive = END_INTERACTIVE
File "/home/student/Harrison/OCaml/interpolation.ml", line 99, characters 6-18:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
((Exists (_, _)|Forall (_, _)|Iff (_, _)|Imp (_, _)|Or (_, _)|Not _|Atom _|
 True|False),
_)
val cinterpolate : fol formula -> fol formula -> fol formula = <fun>
val interpolate : fol formula -> fol formula -> fol formula = <fun>
- : dummy_interactive = START_INTERACTIVE
val p : fol formula =
  <<(forall x. exists y. R(x,y)) /\
    (forall x y. S(v,x,y) <=> R(x,y) \/ R(y,x))>>
val q : fol formula =
  <<(forall x y z. S(v,x,y) /\ S(v,y,z) ==> T(x,z)) /\
    (exists u. ~T(u,u))>>
// interpolation.p009
0 ground instances tried; 0 items in list
0 ground instances tried; 0 items in list
1 ground instances tried; 5 items in list
2 ground instances tried; 6 items in list
3 ground instances tried; 10 items in list
4 ground instances tried; 11 items in list
5 ground instances tried; 16 items in list
6 ground instances tried; 17 items in list
7 ground instances tried; 20 items in list
8 ground instances tried; 21 items in list
8 ground instances tried; 21 items in list
9 ground instances tried; 22 items in list
10 ground instances tried; 23 items in list
11 ground instances tried; 24 items in list
12 ground instances tried; 25 items in list
13 ground instances tried; 29 items in list
14 ground instances tried; 30 items in list
```

```
15 ground instances tried; 34 items in list
16 ground instances tried; 35 items in list
17 ground instances tried; 36 items in list
18 ground instances tried; 37 items in list
19 ground instances tried; 38 items in list
20 ground instances tried; 39 items in list
21 ground instances tried; 43 items in list
22 ground instances tried; 44 items in list
23 ground instances tried; 48 items in list
24 ground instances tried; 49 items in list
25 ground instances tried; 54 items in list
26 ground instances tried; 55 items in list
27 ground instances tried; 59 items in list
28 ground instances tried; 60 items in list
29 ground instances tried; 65 items in list
30 ground instances tried; 66 items in list
val c : fol formula =
  <<forall v_2.
      exists v_1.
        S(v,v_2,v_1) /\ S(v,v_1,v_2) \/ S(v,v_2,v_1) /\ S(v,v_1,v_2)>>
// interpolation.p010
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
- : int list = [4]
// interpolation.p011
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
- : int list = [3]
- : dummy_interactive = END_INTERACTIVE
val einterpolate : fol formula -> fol formula -> fol formula = <fun>
- : dummy_interactive = START_INTERACTIVE
val p : fol formula = <<p ==> q /\ r>>
val q : fol formula = <<~((q ==> p) ==> s ==> (p <=> q))>>
// interpolation.p012
0 ground instances tried; 0 items in list
0 ground instances tried; 0 items in list
val c : fol formula = <<~p \/ ~p \/ q>>
// interpolation.p013
- : bool = true
// interpolation.p014
- : bool = true
// interpolation.p015
- : bool = true
val p : fol formula =
  <<(forall x. exists y. R(x,y)) /\
    (forall x y. S(x,y) <=> R(x,y) \/ R(y,x))>>
val q : fol formula =
  <<(forall x y z. S(x,y) /\ S(y,z) ==> T(x,z)) /\ ~T(u,u)>>
// interpolation.p016
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
- : int list = [5]
// interpolation.p017
0 ground instances tried; 0 items in list
0 ground instances tried; 0 items in list
1 ground instances tried; 5 items in list
1 ground instances tried; 5 items in list
2 ground instances tried; 6 items in list
3 ground instances tried; 10 items in list
val c : fol formula =
  <<forall v_2.
      exists v_1. S(v_2,v_1) /\ S(v_1,v_2) \/ S(v_2,v_1) /\ S(v_1,v_2)>>
// interpolation.p018
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
- : int list = [4]
// interpolation.p019
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
- : int list = [3]
val p : fol formula =
  <<(forall x. exists y. R(x,y)) /\
    (forall x y. S(x,y) <=> R(x,y) \/ R(y,x)) /\
    (forall v. R(u,v) ==> Q(v,u))>>
val q : fol formula =
  <<(forall x y z. S(x,y) /\ S(y,z) ==> T(x,z)) /\ ~T(u,u)>>
// interpolation.p020
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
- : int list = [5]
// interpolation.p021
0 ground instances tried; 0 items in list
0 ground instances tried; 0 items in list
1 ground instances tried; 6 items in list
1 ground instances tried; 6 items in list
2 ground instances tried; 7 items in list
3 ground instances tried; 11 items in list
val c : fol formula =
  <<exists v_1.
      (S(u,v_1) /\ S(v_1,u) \/ S(u,v_1) /\ S(v_1,u)) \/
      (S(u,v_1) /\ S(v_1,u) \/ S(u,v_1) /\ S(v_1,u)) \/
      S(u,v_1) /\ S(v_1,u) \/ S(u,v_1) /\ S(v_1,u)>>
```

```
// interpolation.p022
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
- : int list = [4]
// interpolation.p023
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
- : int list = [3]
File "/home/student/Harrison/OCaml/interpolation.ml", line 197, characters 2-28:
Warning S: this expression should have type unit.
File "/home/student/Harrison/OCaml/interpolation.ml", line 197, characters 30-45:
Warning S: this expression should have type unit.
File "/home/student/Harrison/OCaml/interpolation.ml", line 197, characters 47-66:
Warning S: this expression should have type unit.
val test_interp : fol formula -> fol formula = <fun>
// interpolation.p024
0 ground instances tried; 0 items in list
0 ground instances tried; 0 items in list
1 ground instances tried; 4 items in list
2 ground instances tried; 5 items in list
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
- : fol formula =
<<forall v_2.
    exists v_1.
      (~P(v_2) \/ ~P(v_2) \/ Q(v_1)) \/
      (~P(v_2) \/ ~P(v_2) \/ Q(v_1)) /\ (~P(v_2) \/ Q(v_1))>>
// interpolation.p025
0 ground instances tried; 0 items in list
0 ground instances tried; 0 items in list
1 ground instances tried; 3 items in list
1 ground instances tried; 3 items in list
2 ground instances tried; 6 items in list
3 ground instances tried; 9 items in list
4 ground instances tried; 12 items in list
5 ground instances tried; 14 items in list
6 ground instances tried; 16 items in list
7 ground instances tried; 18 items in list
8 ground instances tried; 21 items in list
9 ground instances tried; 24 items in list
10 ground instances tried; 27 items in list
```

```
10 ground instances tried; 27 items in list
11 ground instances tried; 30 items in list
12 ground instances tried; 33 items in list
13 ground instances tried; 36 items in list
14 ground instances tried; 39 items in list
15 ground instances tried; 42 items in list
16 ground instances tried; 45 items in list
17 ground instances tried; 48 items in list
18 ground instances tried; 51 items in list
19 ground instances tried; 54 items in list
20 ground instances tried; 57 items in list
21 ground instances tried; 60 items in list
22 ground instances tried; 63 items in list
23 ground instances tried; 66 items in list
24 ground instances tried; 69 items in list
25 ground instances tried; 72 items in list
26 ground instances tried; 74 items in list
27 ground instances tried; 76 items in list
28 ground instances tried; 78 items in list
29 ground instances tried; 80 items in list
30 ground instances tried; 82 items in list
31 ground instances tried; 84 items in list
32 ground instances tried; 86 items in list
33 ground instances tried; 88 items in list
34 ground instances tried; 90 items in list
35 ground instances tried; 92 items in list
36 ground instances tried; 94 items in list
37 ground instances tried; 96 items in list
38 ground instances tried; 98 items in list
39 ground instances tried; 100 items in list
40 ground instances tried; 102 items in list
41 ground instances tried; 104 items in list
42 ground instances tried; 106 items in list
43 ground instances tried; 108 items in list
44 ground instances tried; 110 items in list
45 ground instances tried; 112 items in list
46 ground instances tried; 114 items in list
47 ground instances tried; 116 items in list
48 ground instances tried; 118 items in list
49 ground instances tried; 120 items in list
50 ground instances tried; 123 items in list
51 ground instances tried; 126 items in list
52 ground instances tried; 129 items in list
53 ground instances tried; 131 items in list
54 ground instances tried; 133 items in list
55 ground instances tried; 135 items in list
56 ground instances tried; 138 items in list
57 ground instances tried; 141 items in list
58 ground instances tried; 144 items in list
59 ground instances tried; 146 items in list
60 ground instances tried; 148 items in list
61 ground instances tried; 150 items in list
62 ground instances tried; 153 items in list
63 ground instances tried; 156 items in list
64 ground instances tried; 159 items in list
```

```
65 ground instances tried; 161 items in list
66 ground instances tried; 163 items in list
67 ground instances tried; 165 items in list
68 ground instances tried; 168 items in list
69 ground instances tried; 171 items in list
70 ground instances tried; 174 items in list
71 ground instances tried; 177 items in list
72 ground instances tried; 180 items in list
73 ground instances tried; 183 items in list
74 ground instances tried; 186 items in list
75 ground instances tried; 189 items in list
76 ground instances tried; 192 items in list
77 ground instances tried; 195 items in list
78 ground instances tried; 198 items in list
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 0
Searching with depth limit 1
- : fol formula = <<true>>
val p : fol formula = <<forall x. L(x,b)>>
val q : fol formula = <<(forall y. L(b,y) ==> m = y) /\ ~m = b>>
// interpolation.p026
0 ground instances tried; 0 items in list
0 ground instances tried; 0 items in list
val c : fol formula = <<L(b,b)>>
// interpolation.p027
Searching with depth limit 0
Searching with depth limit 1
- : int list = [1]
// interpolation.p028
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
- : int list = [2]
val p : fol formula =
  <<(forall x. A(x) /\ C(x) ==> B(x)) /\
    (forall x. D(x) \/ ~D(x) ==> C(x))>>
val q : fol formula = <<~(forall x. E(x) ==> A(x) ==> B(x))>>
// interpolation.p029
0 ground instances tried; 0 items in list
0 ground instances tried; 0 items in list
val c : fol formula =
  <<forall v_1. (~A(v_1) \/ B(v_1)) \/ ~A(v_1) \/ B(v_1)>>
// interpolation.p030
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
- : int list = [5]
// interpolation.p031
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
```

```
- : int list = [2]
- : dummy_interactive = END_INTERACTIVE
#
```

combining.ml

```
#use combining.ml";;
```

```
# #use "combining.ml";;
val real_lang :
  (string * int -> bool) * (string * int -> bool) *
  (fol formula -> bool) = (<fun>, <fun>, <fun>)
val int_lang :
  (string * int -> bool) * (string * int -> bool) *
  (fol formula -> bool) = (<fun>, <fun>, <fun>)
val add_default :
  (('a -> bool) * (string * int -> bool) * (fol formula -> bool)) list ->
  (('a -> bool) * (string * int -> bool) * (fol formula -> bool)) list =
  <fun>
File "/home/student/Harrison/OCaml/combining.ml", line 42, characters 2-217:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
(Exists (_, _)|Forall (_, _)|Iff (_, _)|Imp (_, _)|Or (_, _)|And (_, _)|
Not _|True|False)
val chooselang :
  ((string * int -> bool) * (string * int -> bool) * 'a) list ->
  fol formula -> (string * int -> bool) * (string * int -> bool) * 'a =
  <fun>
val listify :
  ('a -> ('b -> 'c) -> 'c) -> 'a list -> ('b list -> 'c) -> 'c = <fun>
val homot :
  (string * int -> bool) * 'a * 'b ->
  term ->
  (term -> num -> fol formula list -> 'c) ->
  num -> fol formula list -> 'c = <fun>
val homol :
  ((string * int -> bool) * (string * int -> bool) * 'a) list ->
  fol formula ->
  (fol formula -> num -> fol formula list -> 'b) ->
  num -> fol formula list -> 'b = <fun>
val homo :
  ((string * int -> bool) * (string * int -> bool) * 'a) list ->
  fol formula list ->
  (fol formula list -> num -> fol formula list -> 'b) ->
  num -> fol formula list -> 'b = <fun>
val homogenize :
  ((string * int -> bool) * (string * int -> bool) * 'a) list ->
  fol formula list -> fol formula list = <fun>
val belongs :
  (string * int -> bool) * (string * int -> bool) * 'a ->
  fol formula -> bool = <fun>
val langpartition :
  ((string * int -> bool) * (string * int -> bool) * 'a) list ->
  fol formula list -> fol formula list list = <fun>
- : dummy_interactive = START_INTERACTIVE
// combining.p001
- : fol formula = <<true>>
```

```
// combining.p002
- : bool = true
- : dummy_interactive = END_INTERACTIVE
val arreq : string list -> fol formula list = <fun>
val arrangement : string list list -> fol formula list = <fun>
val dest_def : fol formula -> string * term = <fun>
val redeqs : fol formula list -> fol formula list = <fun>
val trydps :
  (('a * 'b * (fol formula -> bool)) * fol formula list) list ->
  fol formula list -> bool = <fun>
val allpartitions : 'a list -> 'a list list list = <fun>
val nelop_refute :
  string list ->
  (('a * 'b * (fol formula -> bool)) * fol formula list) list -> bool =
  <fun>
val nelop1 :
  ((string * int -> bool) * (string * int -> bool) *
   (fol formula -> bool))
  list -> fol formula list -> bool = <fun>
val nelop :
  ((string * int -> bool) * (string * int -> bool) *
   (fol formula -> bool))
  list -> fol formula -> bool = <fun>
- : dummy_interactive = START_INTERACTIVE
// combining.p003
- : bool = true
// combining.p004
- : int list = [1; 2; 5; 15; 52; 203; 877; 4140; 21147; 115975]
- : dummy_interactive = END_INTERACTIVE
val findasubset : ('a list -> 'b) -> int -> 'a list -> 'b = <fun>
val findsubset : ('a list -> bool) -> 'a list -> 'a list = <fun>
val nelop_refute :
  fol formula list ->
  (('a * 'b * (fol formula -> bool)) * fol formula list) list -> bool =
  <fun>
val nelop1 :
  ((string * int -> bool) * (string * int -> bool) *
   (fol formula -> bool))
  list -> fol formula list -> bool = <fun>
val nelop :
  ((string * int -> bool) * (string * int -> bool) *
   (fol formula -> bool))
  list -> fol formula -> bool = <fun>
- : dummy_interactive = START_INTERACTIVE
// combining.p005
- : bool = true
// combining.p006
- : bool = true
// combining.p007
- : bool = true
// combining.p008
- : fol formula list = [<<true>>; <<false>>; <<false>>]
// combining.p009
- : fol formula list = [<<true>>; <<false>>; <<false>>]
// combining.p010
```

```
- : bool = true
// combining.p011
- : bool = true
// combining.p012
CPU time (user): 0.00400000000002
- : bool = true
// combining.p013
CPU time (user): 0.0120010000001
- : bool = false
// combining.p014
CPU time (user): 0.
- : bool = true
// combining.p015
CPU time (user): 0.
- : bool = false
// combining.p016
CPU time (user): 0.032002
- : bool = true
// combining.p018
CPU time (user): 0.
- : bool = true
// combining.p019
CPU time (user): 0.
- : bool = true
// combining.p020
CPU time (user): 0.0120009999999
- : bool = true
// combining.p021
CPU time (user): 0.00400100000002
- : bool = true
// combining.p023
- : bool = true
- : dummy_interactive = END_INTERACTIVE
#
```

lcf.ml

```
#use "lcf.ml";;
```

```
# #use "lcf.ml";;
module type Proofsystem =
  sig
    type thm
    val modusponens : thm -> thm -> thm
    val gen : string -> thm -> thm
    val axiom_addimp : fol formula -> fol formula -> thm
    val axiom_distribimp :
      fol formula -> fol formula -> fol formula -> thm
    val axiom_doubleneg : fol formula -> thm
    val axiom_allimp : string -> fol formula -> fol formula -> thm
    val axiom_impall : string -> fol formula -> thm
    val axiom_existseq : string -> term -> thm
    val axiom_eqrefl : term -> thm
    val axiom_funcong : string -> term list -> term list -> thm
    val axiom_predcong : string -> term list -> term list -> thm
    val axiom_iffimp1 : fol formula -> fol formula -> thm
    val axiom_iffimp2 : fol formula -> fol formula -> thm
    val axiom_impiff : fol formula -> fol formula -> thm
    val axiom_true : thm
    val axiom_not : fol formula -> thm
    val axiom_and : fol formula -> fol formula -> thm
    val axiom_or : fol formula -> fol formula -> thm
    val axiom_exists : string -> fol formula -> thm
    val concl : thm -> fol formula
  end
val occurs_in : term -> term -> bool = <fun>
val free_in : term -> fol formula -> bool = <fun>
module Proven : Proofsystem
type thm = Proven.thm
val modusponens : thm -> thm -> thm = <fun>
val gen : string -> thm -> thm = <fun>
val axiom_addimp : fol formula -> fol formula -> thm = <fun>
val axiom_distribimp : fol formula -> fol formula -> fol formula -> thm =
  <fun>
val axiom_doubleneg : fol formula -> thm = <fun>
val axiom_allimp : string -> fol formula -> fol formula -> thm = <fun>
val axiom_impall : string -> fol formula -> thm = <fun>
val axiom_existseq : string -> term -> thm = <fun>
val axiom_eqrefl : term -> thm = <fun>
val axiom_funcong : string -> term list -> term list -> thm = <fun>
val axiom_predcong : string -> term list -> term list -> thm = <fun>
val axiom_iffimp1 : fol formula -> fol formula -> thm = <fun>
val axiom_iffimp2 : fol formula -> fol formula -> thm = <fun>
val axiom_impiff : fol formula -> fol formula -> thm = <fun>
val axiom_true : thm = <abstr>
val axiom_not : fol formula -> thm = <fun>
val axiom_and : fol formula -> fol formula -> thm = <fun>
val axiom_or : fol formula -> fol formula -> thm = <fun>
val axiom_exists : string -> fol formula -> thm = <fun>
val concl : thm -> fol formula = <fun>
val print_thm : thm -> unit = <fun>
#
```

lcfprop.ml

```
#use "lcfprop.ml";;
```

```
# #use "lcfprop.ml";;
val imp_refl : fol formula -> thm = <fun>
val imp_unduplicate : thm -> thm = <fun>
val negatef : 'a formula -> 'a formula = <fun>
val negativef : 'a formula -> bool = <fun>
val add_assum : fol formula -> thm -> thm = <fun>
val imp_add_assum : fol formula -> thm -> thm = <fun>
val imp_trans : thm -> thm -> thm = <fun>
val imp_insert : fol formula -> thm -> thm = <fun>
val imp_swap : thm -> thm = <fun>
val imp_trans_th : fol formula -> fol formula -> fol formula -> thm =
  <fun>
val imp_add_concl : fol formula -> thm -> thm = <fun>
val imp_swap_th : fol formula -> fol formula -> fol formula -> thm =
  <fun>
val imp_swap2 : thm -> thm = <fun>
val right_mp : thm -> thm -> thm = <fun>
val iff_imp1 : thm -> thm = <fun>
val iff_imp2 : thm -> thm = <fun>
val imp_antisym : thm -> thm -> thm = <fun>
val right_doubleneg : thm -> thm = <fun>
val ex_falso : fol formula -> thm = <fun>
File "/home/student/Harrison/OCaml/lcfprop.ml", line 189, characters 6-21:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
(Exists (_, _)|Forall (_, _)|Iff (_, _)|Or (_, _)|And (_, _)|Not _|Atom _|
True|False)
File "/home/student/Harrison/OCaml/lcfprop.ml", line 189, characters 38-47:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
(Exists (_, _)|Forall (_, _)|Iff (_, _)|Or (_, _)|And (_, _)|Not _|Atom _|
True|False)
val imp_trans2 : thm -> thm -> thm = <fun>
val imp_trans_chain : thm list -> thm -> thm = <fun>
val imp_truefalse : fol formula -> fol formula -> thm = <fun>
val imp_mono_th :
  fol formula -> fol formula -> fol formula -> fol formula -> thm =
  <fun>
val truth : thm = |- true
val contrapos : thm -> thm = <fun>
val and_left : fol formula -> fol formula -> thm = <fun>
val and_right : fol formula -> fol formula -> thm = <fun>
val conjths : fol formula -> thm list = <fun>
val and_pair : fol formula -> fol formula -> thm = <fun>
val shunt : thm -> thm = <fun>
val unshunt : thm -> thm = <fun>
val iff_def : fol formula -> fol formula -> thm = <fun>
val iff_def : fol formula -> fol formula -> thm = <fun>
val expand_connective : fol formula -> thm = <fun>
val eliminate_connective : fol formula -> thm = <fun>
val imp_false_conseqs : fol formula -> fol formula -> thm list = <fun>
val imp_false_rule : thm -> thm = <fun>
val imp_true_rule : thm -> thm -> thm = <fun>
val imp_contr : fol formula -> fol formula -> thm = <fun>
val imp_front_th : int -> fol formula -> thm = <fun>
val imp_front : int -> thm -> thm = <fun>
val lcfptab : fol formula list -> fol formula list -> thm = <fun>
val lcftaut : fol formula -> thm = <fun>
- : dummy_interactive = START_INTERACTIVE
// lcfprop.p001
- : thm = |- (p ==> q) \/ (q ==> p)
// lcfprop.p002
- : thm = |- p /\ q <=> (p <=> q) <=> p \/ q
// lcfprop.p003
- : thm = |- ((p <=> q) <=> r) <=> p <=> q <=> r
- : dummy_interactive = END_INTERACTIVE
#
```

folderived.ml

```
#use "folderived.ml";;
```

```
# #use "folderived.ml";;
val eq_sym : term -> term -> thm = <fun>
val eq_trans : term -> term -> term -> thm = <fun>
val icongruence : term -> term -> term -> term -> thm = <fun>
- : dummy_interactive = START_INTERACTIVE
// folderived.p001
- : thm =
|- s = t ==> f(s,g(s,t,s),u,h(h(s))) = f(s,g(t,t,s),u,h(h(t)))
- : dummy_interactive = END_INTERACTIVE
val gen_right_th : string -> fol formula -> fol formula -> thm = <fun>
val genimp : string -> thm -> thm = <fun>
val gen_right : string -> thm -> thm = <fun>
val exists_left_th : string -> fol formula -> fol formula -> thm =
  <fun>
val exists_left : string -> thm -> thm = <fun>
val subspec : thm -> thm = <fun>
val subalpha : thm -> thm = <fun>
val isubst : term -> term -> fol formula -> fol formula -> thm = <fun>
val alpha : string -> fol formula -> thm = <fun>
val ispec : term -> fol formula -> thm = <fun>
val spec : term -> thm -> thm = <fun>
- : dummy_interactive = START_INTERACTIVE
// folderived.p002
- : thm =
|-
(forall x y z. x + y + z = z + y + x) ==>
(forall y' z. y + y' + z = z + y' + y)
// folderived.p003
- : thm =
|- x + x = 2 * x ==> (x + x = x ==> x = 0) ==> 2 * x = x ==> x = 0
// folderived.p004
- : thm =
|-
x + x = 2 * x ==>
(x + x = y + y ==> y + y + y = x + x + x) ==>
2 * x = y + y ==> y + y + y = x + 2 * x
// folderived.p005
- : thm =
|-
(forall x y z. x + y + z = y + z + z) ==>
(forall y z. x + y + z = y + z + z)
// folderived.p006
- : thm = |- (forall x. x = x) ==> x = x
// folderived.p007
- : thm =
|-
(forall x y z. x + y + z = y + z + z) ==>
```

```
(forall y' z'. (w + y + z) + y' + z' = y' + z' + z')
// folderived.p008
- : thm =
|-
(forall x y z. x + y + z = y + z + z) ==>
(forall y' z'. (x + y + z) + y' + z' = y' + z' + z')
// folderived.p009
- : thm =
|- (forall x y z. nothing_much) ==> (forall y z. nothing_much)
// folderived.p010
- : fol formula -> thm = <fun>
// folderived.p011
- : thm =
|-
x + x = 2 * x ==>
((exists x. x = 2) <=> (exists y. y + x + x = y + y + y)) ==>
((exists x. x = 2) <=> (exists y. y + 2 * x = y + y + y))
// folderived.p012
- : thm =
|-
x = y ==>
((forall z. x = z) <=> (exists x. y < z) /\ (forall y. y < x)) ==>
((forall z. y = z) <=> (exists x. y < z) /\ (forall y'. y' < y))
// folderived.p013
- : thm =
|-
(forall x x' x''. x + x' + x'' = 0) ==>
(forall x'' x'''. x' + x'' + x''' = 0)
// folderived.p014
- : thm =
|-
(forall x x' x''. x + x' + x'' = 0) ==>
(forall x' x'''. x'' + x' + x''' = 0)
// folderived.p015
- : thm =
|-
(forall x x' x''. x + x' + x'' = 0) ==>
(forall x''' x''''. (x' + x'') + x''' + x'''' = 0)
// folderived.p016
- : thm =
|-
(forall x x' x''. x + x' + x'' = 0) ==>
(forall x''' x''''. (x + x' + x'') + x''' + x'''' = 0)
// folderived.p017
- : thm =
|-
(forall x x'. x + x' = x' + x) ==> (forall x'. 2 * x + x' = x' + 2 * x)
- : dummy_interactive = END_INTERACTIVE
#
```

lcffol.ml

```
#use "lcffol.ml";;
```

```
# #use "lcffol.ml";;
val unify_complementsf :
  (string, term) func ->
  fol formula * fol formula -> (string, term) func = <fun>
File "/home/student/Harrison/OCaml/lcffol.ml", line 24, characters 2-402:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
(Exists (_, _)|Forall (_, _)|Iff (_, _)|Or (_, _)|And (_, _)|Not _|Atom _|
True|False)
val use_laterimp : fol formula -> fol formula -> thm = <fun>
val imp_false_rule' : ('a -> thm) -> 'a -> thm = <fun>
val imp_true_rule' : ('a -> thm) -> ('a -> thm) -> 'a -> thm = <fun>
val imp_front' : int -> ('a -> thm) -> 'a -> thm = <fun>
val add_assum' :
  fol formula ->
  ((term -> term) * 'a -> thm) -> (term -> term) * 'a -> thm = <fun>
val eliminate_connective' :
  fol formula ->
  ((term -> term) * 'a -> thm) -> (term -> term) * 'a -> thm = <fun>
val spec' :
  term ->
  fol formula ->
  int -> ((term -> term) * 'a -> thm) -> (term -> term) * 'a -> thm =
  <fun>
val ex_falso' : fol formula list -> (term -> term) * fol formula -> thm =
  <fun>
File "/home/student/Harrison/OCaml/lcffol.ml", line 57, characters 6-15:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
(_, [])
File "/home/student/Harrison/OCaml/lcffol.ml", line 56, characters 14-216:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
([], _)
val complits' :
  fol formula list * fol formula list ->
  int -> (term -> term) * fol formula -> thm = <fun>
val deskol' :
  fol formula ->
  ((term -> term) * 'a -> thm) -> (term -> term) * 'a -> thm = <fun>
val lcftab :
  (fol formula -> term) ->
  fol formula list * fol formula list * int ->
  (((term -> term) * fol formula -> thm) ->
   (string, term) func * (fol formula * term) list * int -> 'a) ->
  (string, term) func * (fol formula * term) list * int -> 'a = <fun>
val quantforms : bool -> 'a formula -> 'a formula list = <fun>
```

```
File "/home/student/Harrison/OCaml/lcffol.ml", line 131, characters 15-136:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
(Exists (_, _)|Iff (_, _)|Imp (_, _)|Or (_, _)|And (_, _)|Not _|Atom _|True|
False)
val skolemfuns : fol formula -> (fol formula * term) list = <fun>
val form_match :
  fol formula * fol formula ->
  (string, term) func -> (string, term) func = <fun>
val lcfrefute :
  fol formula ->
  int ->
  (((term -> term) * fol formula -> thm) ->
   (string, term) func * (fol formula * term) list * int -> 'a) ->
  'a = <fun>
File "/home/student/Harrison/OCaml/lcffol.ml", line 168, characters 12-77:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
((Exists (_, _)|Iff (_, _)|Imp (_, _)|Or (_, _)|And (_, _)|Not _|Atom _|True|
 False),
_)
val mk_skol : fol formula * term -> fol formula -> fol formula = <fun>
val simpcont :
  ((term -> term) * fol formula -> 'a) ->
  (string, term) func * (fol formula * term) list * 'b -> 'a = <fun>
- : thm = |- p(1) /\ ~q(1) /\ (forall x. p(x) ==> q(x)) ==> false
- : thm =
|-
(exists x. ~p(x)) /\ (forall x. p(x)) ==>
(~(~p(f_1)) ==> (forall x. ~(~p(x)))) ==> false
File "/home/student/Harrison/OCaml/lcffol.ml", line 188, characters 12-21:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
[]
val elim_skolemvar : thm -> thm = <fun>
val deskolcont :
  ((term -> term) * fol formula -> thm) ->
  (string, term) func * (fol formula * term) list * 'a -> thm = <fun>
val lcffol : fol formula -> thm = <fun>
- : dummy_interactive = START_INTERACTIVE
// lcffol.p003
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
val p58 : thm =
  |-
  forall x.
    exists v w.
      forall y z. P(x) /\ Q(y) ==> (P(v) \/ R(w)) /\ (R(z) ==> Q(v))
// lcffol.p004
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
val ewd1062_1 : thm =
  |-
  (forall x. x <= x) /\
  (forall x y z. x <= y /\ y <= z ==> x <= z) /\
  (forall x y. f(x) <= y <=> x <= g(y)) ==>
  (forall x y. x <= y ==> f(x) <= f(y))
- : dummy_interactive = END_INTERACTIVE
- : dummy_interactive = START_INTERACTIVE
val start_time : float = 585.196572
// lcffol.p005
CPU time (user): 0.
val p1 : thm = |- p ==> q <=> ~q ==> ~p
// lcffol.p006
CPU time (user): 0.
val p2 : thm = |- ~(~p) <=> p
// lcffol.p007
CPU time (user): 0.
val p3 : thm = |- ~(p ==> q) ==> q ==> p
// lcffol.p008
CPU time (user): 0.00400000000002
val p4 : thm = |- ~p ==> q <=> ~q ==> p
// lcffol.p009
CPU time (user): 0.
val p5 : thm = |- (p \/ q ==> p \/ r) ==> p \/ (q ==> r)
// lcffol.p010
CPU time (user): 0.
val p6 : thm = |- p \/ ~p
// lcffol.p011
CPU time (user): 0.
val p7 : thm = |- p \/ ~(~(~p))
// lcffol.p012
CPU time (user): 0.
val p8 : thm = |- ((p ==> q) ==> p) ==> p
// lcffol.p013
CPU time (user): 0.00400000000002
val p9 : thm = |- (p \/ q) /\ (~p \/ q) /\ (p \/ ~q) ==> ~(~q \/ ~q)
// lcffol.p014
CPU time (user): 0.
val p10 : thm =
  |- (q ==> r) /\ (r ==> p /\ q) /\ (p ==> q /\ r) ==> (p <=> q)
// lcffol.p015
CPU time (user): 0.
val p11 : thm = |- p <=> p
// lcffol.p016
CPU time (user): 0.016001
val p12 : thm = |- ((p <=> q) <=> r) <=> p <=> q <=> r
// lcffol.p017
CPU time (user): 0.00400100000002
val p13 : thm = |- p \/ q /\ r <=> (p \/ q) /\ (p \/ r)
// lcffol.p018
CPU time (user): 0.00400000000002
```

```
val p14 : thm = |- (p <=> q) <=> (q \/ ~p) /\ (~q \/ p)
// lcffol.p019
CPU time (user): 0.
val p15 : thm = |- p ==> q <=> ~p \/ q
// lcffol.p020
CPU time (user): 0.
val p16 : thm = |- (p ==> q) \/ (q ==> p)
// lcffol.p021
CPU time (user): 0.00800100000004
val p17 : thm =
  |- p /\ (q ==> r) ==> s <=> (~p \/ q \/ s) /\ (~p \/ ~r \/ s)
// lcffol.p022
Searching with depth limit 0
CPU time (user): 0.
val p1 : thm = |- p ==> q <=> ~q ==> ~p
// lcffol.p023
Searching with depth limit 0
CPU time (user): 0.
val p2 : thm = |- ~(~p) <=> p
// lcffol.p024
Searching with depth limit 0
CPU time (user): 0.
val p3 : thm = |- ~(p ==> q) ==> q ==> p
// lcffol.p025
Searching with depth limit 0
CPU time (user): 0.00400000000002
val p4 : thm = |- ~p ==> q <=> ~q ==> p
// lcffol.p026
Searching with depth limit 0
CPU time (user): 0.
val p5 : thm = |- (p \/ q ==> p \/ r) ==> p \/ (q ==> r)
// lcffol.p027
Searching with depth limit 0
CPU time (user): 0.
val p6 : thm = |- p \/ ~p
// lcffol.p028
Searching with depth limit 0
CPU time (user): 0.
val p7 : thm = |- p \/ ~(~(~p))
// lcffol.p029
Searching with depth limit 0
CPU time (user): 0.
val p8 : thm = |- ((p ==> q) ==> p) ==> p
// lcffol.p030
Searching with depth limit 0
CPU time (user): 0.
val p9 : thm = |- (p \/ q) /\ (~p \/ q) /\ (p \/ ~q) ==> ~(~q \/ ~q)
// lcffol.p031
Searching with depth limit 0
CPU time (user): 0.
val p10 : thm =
  |- (q ==> r) /\ (r ==> p /\ q) /\ (p ==> q /\ r) ==> (p <=> q)
// lcffol.p032
Searching with depth limit 0
CPU time (user): 0.
```

```
val p11 : thm = |- p <=> p
// lcffol.p033
Searching with depth limit 0
CPU time (user): 0.0160010000001
val p12 : thm = |- ((p <=> q) <=> r) <=> p <=> q <=> r
// lcffol.p034
Searching with depth limit 0
CPU time (user): 0.00399999999991
val p13 : thm = |- p \/ q /\ r <=> (p \/ q) /\ (p \/ r)
// lcffol.p035
Searching with depth limit 0
CPU time (user): 0.00400100000002
val p14 : thm = |- (p <=> q) <=> (q \/ ~p) /\ (~q \/ p)
// lcffol.p036
Searching with depth limit 0
CPU time (user): 0.
val p15 : thm = |- p ==> q <=> ~p \/ q
// lcffol.p037
Searching with depth limit 0
CPU time (user): 0.
val p16 : thm = |- (p ==> q) \/ (q ==> p)
// lcffol.p038
Searching with depth limit 0
CPU time (user): 0.00800000000004
val p17 : thm =
  |- p /\ (q ==> r) ==> s <=> (~p \/ q \/ s) /\ (~p \/ ~r \/ s)
// lcffol.p039
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
CPU time (user): 0.
val p18 : thm = |- exists y. forall x. P(y) ==> P(x)
// lcffol.p040
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
CPU time (user): 0.0320019999999
val p19 : thm =
  |- exists x. forall y z. (P(y) ==> Q(z)) ==> P(x) ==> Q(x)
// lcffol.p041
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
CPU time (user): 0.024001
val p20 : thm =
  |-
  (forall x y. exists z. forall w. P(x) /\ Q(y) ==> R(z) /\ U(w)) ==>
  (exists x y. P(x) /\ Q(y)) ==> (exists z. R(z))
// lcffol.p042
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
CPU time (user): 0.0120010000001
```

```
val p21 : thm =
  |-
  (exists x. P ==> Q(x)) /\ (exists x. Q(x) ==> P) ==>
  (exists x. P <=> Q(x))
// lcffol.p043
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
CPU time (user): 0.00399999999991
val p22 : thm = |- (forall x. P <=> Q(x)) ==> (P <=> (forall x. Q(x)))
// lcffol.p044
Searching with depth limit 0
Searching with depth limit 1
CPU time (user): 0.00400100000002
val p23 : thm = |- (forall x. P \/ Q(x)) <=> P \/ (forall x. Q(x))
// lcffol.p045
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
CPU time (user): 0.028002
val p24 : thm =
  |-
  ~(exists x. U(x) /\ Q(x)) /\
  (forall x. P(x) ==> Q(x) \/ R(x)) /\
  ~(exists x. P(x) ==> (exists x. Q(x))) /\
  (forall x. Q(x) /\ R(x) ==> U(x)) ==> (exists x. P(x) /\ R(x))
// lcffol.p046
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
CPU time (user): 0.00800000000004
val p25 : thm =
  |-
  (exists x. P(x)) /\
  (forall x. U(x) ==> ~G(x) /\ R(x)) /\
  (forall x. P(x) ==> G(x) /\ U(x)) /\
  ((forall x. P(x) ==> Q(x)) \/ (exists x. Q(x) /\ P(x))) ==>
  (exists x. Q(x) /\ P(x))
// lcffol.p047
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
CPU time (user): 0.164011
val p26 : thm =
  |-
  ((exists x. P(x)) <=> (exists x. Q(x))) /\
  (forall x y. P(x) /\ Q(y) ==> (R(x) <=> U(y))) ==>
  ((forall x. P(x) ==> R(x)) <=> (forall x. Q(x) ==> U(x)))
```

```
// lcffol.p048
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
CPU time (user): 0.0560029999999
val p27 : thm =
  |-
  (exists x. P(x) /\ ~Q(x)) /\
  (forall x. P(x) ==> R(x)) /\
  (forall x. U(x) /\ V(x) ==> P(x)) /\ (exists x. R(x) /\ ~Q(x)) ==>
  (forall x. U(x) ==> ~R(x)) ==> (forall x. U(x) ==> ~V(x))
// lcffol.p049
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
CPU time (user): 0.044002
val p28 : thm =
  |-
  (forall x. P(x) ==> (forall x. Q(x))) /\
  ((forall x. Q(x) \/ R(x)) ==> (exists x. Q(x) /\ R(x))) /\
  ((exists x. R(x)) ==> (forall x. L(x) ==> M(x))) ==>
  (forall x. P(x) /\ L(x) ==> M(x))
// lcffol.p050
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
CPU time (user): 0.0600030000001
val p29 : thm =
  |-
  (exists x. P(x)) /\ (exists x. G(x)) ==>
  ((forall x. P(x) ==> H(x)) /\ (forall x. G(x) ==> J(x)) <=>
   (forall x y. P(x) /\ G(y) ==> H(x) /\ J(y)))
// lcffol.p051
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
CPU time (user): 0.00400100000002
val p30 : thm =
  |-
  (forall x. P(x) \/ G(x) ==> ~H(x)) /\
  (forall x. (G(x) ==> ~U(x)) ==> P(x) /\ H(x)) ==> (forall x. U(x))
// lcffol.p052
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
CPU time (user): 0.00799999999992
val p31 : thm =
  |-
  ~(exists x. P(x) /\ (G(x) \/ H(x))) /\
  (exists x. Q(x) /\ P(x)) /\ (forall x. ~H(x) ==> J(x)) ==>
  (exists x. Q(x) /\ J(x))
// lcffol.p053
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
CPU time (user): 0.0360020000001
val p32 : thm =
  |-
  (forall x. P(x) /\ (G(x) \/ H(x)) ==> Q(x)) /\
  (forall x. Q(x) /\ H(x) ==> J(x)) /\ (forall x. R(x) ==> H(x)) ==>
  (forall x. P(x) /\ R(x) ==> J(x))
// lcffol.p054
Searching with depth limit 0
Searching with depth limit 1
CPU time (user): 0.140009
val p33 : thm =
  |-
  (forall x. P(a) /\ (P(x) ==> P(b)) ==> P(c)) <=>
  (forall x. P(a) ==> P(x) \/ P(c)) /\ (P(a) ==> P(b) ==> P(c))
// lcffol.p056
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
CPU time (user): 0.00400000000002
val p35 : thm = |- exists x y. P(x,y) ==> (forall x y. P(x,y))
// lcffol.p057
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
CPU time (user): 0.0200020000001
val p36 : thm =
  |-
  (forall x. exists y. P(x,y)) /\
  (forall x. exists y. G(x,y)) /\
  (forall x y.
     P(x,y) \/ G(x,y) ==> (forall z. P(y,z) \/ G(y,z) ==> H(x,z))) ==>
  (forall x. exists y. H(x,y))
// lcffol.p058
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
CPU time (user): 0.16401
val p37 : thm =
  |-
  (forall z.
     exists w.
```

```
       forall x.
         exists y.
           (P(x,z) ==> P(y,w)) /\
           P(y,z) /\ (P(y,w) ==> (exists u. Q(u,w)))) /\
  (forall x z. ~P(x,z) ==> (exists y. Q(y,z))) /\
  ((exists x y. Q(x,y)) ==> (forall x. R(x,x))) ==>
  (forall x. exists y. R(x,y))
// lcffol.p059
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
CPU time (user): 1.12007
val p38 : thm =
  |-
  (forall x.
     P(a) /\ (P(x) ==> (exists y. P(y) /\ R(x,y))) ==>
     (exists z w. P(z) /\ R(x,w) /\ R(w,z))) <=>
  (forall x.
     (~P(a) \/ P(x) \/ (exists z w. P(z) /\ R(x,w) /\ R(w,z))) /\
     (~P(a) \/
      ~(exists y. P(y) /\ R(x,y)) \/
      (exists z w. P(z) /\ R(x,w) /\ R(w,z))))
// lcffol.p060
Searching with depth limit 0
Searching with depth limit 1
CPU time (user): 0.00400000000002
val p39 : thm = |- ~(exists x. forall y. P(y,x) <=> ~P(y,y))
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
// lcffol.p061
CPU time (user): 0.020001
val p40 : thm =
  |-
  (exists y. forall x. P(x,y) <=> P(x,x)) ==>
  ~(forall x. exists y. forall z. P(z,y) <=> ~P(z,x))
// lcffol.p062
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
CPU time (user): 0.016001
val p41 : thm =
  |-
  (forall z. exists y. forall x. P(x,y) <=> P(x,z) /\ ~P(x,x)) ==>
  ~(exists z. forall x. P(x,z))
// lcffol.p063
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
```

```
CPU time (user): 0.052003
val p42 : thm =
  |- ~(exists y. forall x. P(x,y) <=> ~(exists z. P(x,z) /\ P(z,x)))
// lcffol.p065
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
CPU time (user): 0.028002
val p44 : thm =
  |-
  (forall x.
     P(x) ==> (exists y. G(y) /\ H(x,y)) /\ (exists y. G(y) /\ ~H(x,y))) /\
  (exists x. J(x) /\ (forall y. G(y) ==> H(x,y))) ==>
  (exists x. J(x) /\ ~P(x))
// lcffol.p067
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
Searching with depth limit 8
CPU time (user): 6.756422
val p55 : thm =
  |-
  lives(agatha) /\
  lives(butler) /\
  lives(charles) /\
  (killed(agatha,agatha) \/
   killed(butler,agatha) \/ killed(charles,agatha)) /\
  (forall x y. killed(x,y) ==> hates(x,y) /\ ~richer(x,y)) /\
  (forall x. hates(agatha,x) ==> ~hates(charles,x)) /\
  (hates(agatha,agatha) /\ hates(agatha,charles)) /\
  (forall x. lives(x) /\ ~richer(x,agatha) ==> hates(butler,x)) /\
  (forall x. hates(agatha,x) ==> hates(butler,x)) /\
  (forall x. ~hates(x,agatha) \/ ~hates(x,butler) \/ ~hates(x,charles)) ==>
  killed(agatha,agatha) /\
  ~killed(butler,agatha) /\ ~killed(charles,agatha)
// lcffol.p068
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
CPU time (user): 0.040003
val p57 : thm =
  |-
  P(f(a,b),f(b,c)) /\
  P(f(b,c),f(a,c)) /\ (forall x y z. P(x,y) /\ P(y,z) ==> P(x,z)) ==>
  P(f(a,b),f(a,c))
// lcffol.p069
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
CPU time (user): 0.104007
val p58 : thm =
  |-
  forall P Q R x.
    exists v w.
      forall y z. P(x) /\ Q(y) ==> (P(v) \/ R(w)) /\ (R(z) ==> Q(v))
// lcffol.p070
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
CPU time (user): 0.00799999999992
val p59 : thm =
  |- (forall x. P(x) <=> ~P(f(x))) ==> (exists x. P(x) /\ ~P(f(x)))
// lcffol.p071
Searching with depth limit 0
Searching with depth limit 1
CPU time (user): 0.0200010000001
val p60 : thm =
  |-
  forall x.
    P(x,f(x)) <=>
    (exists y. (forall z. P(z,y) ==> P(z,f(x))) /\ P(x,y))
// lcffol.p072
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
CPU time (user): 0.696044
val gilmore_3 : thm =
  |-
  exists x.
    forall y z.
      ((F(y,z) ==> G(y) ==> H(x)) ==> F(x,x)) /\
      ((F(z,x) ==> G(x)) ==> H(z)) /\ F(x,y) ==> F(z,z)
// lcffol.p073
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
CPU time (user): 2.592162
val gilmore_4 : thm =
  |-
  exists x y.
    forall z.
      (F(x,y) ==> F(y,z) /\ F(z,z)) /\
      (F(x,y) /\ G(x,y) ==> G(x,z) /\ G(z,z))
// lcffol.p074
```

```
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
CPU time (user): 0.00800100000004
val gilmore_5 : thm =
  |-
  (forall x. exists y. F(x,y) \/ F(y,x)) /\
  (forall x y. F(y,x) ==> F(y,y)) ==> (exists z. F(z,z))
// lcffol.p075
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
CPU time (user): 0.048003
val gilmore_6 : thm =
  |-
  forall x.
    exists y.
      (exists u. forall v. F(u,x) ==> G(v,u) /\ G(u,x)) ==>
      (exists u. forall v. F(u,y) ==> G(v,u) /\ G(u,y)) \/
      (forall u v. exists w. G(v,u) \/ H(w,y,u) ==> G(u,w))
// lcffol.p076
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
CPU time (user): 0.020001
val gilmore_7 : thm =
  |-
  (forall x. K(x) ==> (exists y. L(y) /\ (F(x,y) ==> G(x,y)))) /\
  (exists z. K(z) /\ (forall u. L(u) ==> F(z,u))) ==>
  (exists v w. K(v) /\ L(w) /\ G(v,w))
// lcffol.p077
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
CPU time (user): 2.956185
val gilmore_8 : thm =
  |-
  exists x.
    forall y z.
      ((F(y,z) ==> G(y) ==> (forall u. exists v. H(u,v,x))) ==> F(x,x)) /\
      ((F(z,x) ==> G(x)) ==> (forall u. exists v. H(u,v,z))) /\ F(x,y) ==>
      F(z,z)
// lcffol.p078
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
CPU time (user): 3.620226
val gilmore_9 : thm =
  |-
  forall x.
    exists y.
      forall z.
        ((forall u. exists v. F(y,u,v) /\ G(y,u) /\ ~H(y,x)) ==>
         (forall u. exists v. F(x,u,v) /\ G(z,u) /\ ~H(x,z)) ==>
         (forall u. exists v. F(x,u,v) /\ G(y,u) /\ ~H(x,y))) /\
        ((forall u. exists v. F(x,u,v) /\ G(y,u) /\ ~H(x,y)) ==>
         ~(forall u. exists v. F(x,u,v) /\ G(z,u) /\ ~H(x,z)) ==>
         (forall u. exists v. F(y,u,v) /\ G(y,u) /\ ~H(y,x)) /\
         (forall u. exists v. F(z,u,v) /\ G(y,u) /\ ~H(z,y)))
// lcffol.p079
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
CPU time (user): 2.588161
val davis_putnam_example : thm =
  |-
  exists x y.
    forall z.
      (F(x,y) ==> F(y,z) /\ F(z,z)) /\
      (F(x,y) /\ G(x,y) ==> G(x,z) /\ G(z,z))
// lcffol.p080
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
CPU time (user): 0.064004
val ewd1062_1 : thm =
  |-
  (forall x. x <= x) /\
  (forall x y z. x <= y /\ y <= z ==> x <= z) /\
  (forall x y. f(x) <= y <=> x <= g(y)) ==>
  (forall x y. x <= y ==> f(x) <= f(y))
// lcffol.p081
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
CPU time (user): 0.084005
val ewd1062_2 : thm =
  |-
  (forall x. x <= x) /\
  (forall x y z. x <= y /\ y <= z ==> x <= z) /\
  (forall x y. f(x) <= y <=> x <= g(y)) ==>
  (forall x y. x <= y ==> g(x) <= g(y))
Complete CPU time (user): 21.849365- : unit = ()

- : unit = ()
- : dummy_interactive = END_INTERACTIVE
#
```

tactics.ml

```
#use tactics.ml";;
```

```
# #use "tactics.ml";;
type goals =
    Goals of ((string * fol formula) list * fol formula) list *
      (thm list -> thm)
val print_goal : goals -> unit = <fun>
File "/home/student/Harrison/OCaml/tactics.ml", line 41, characters 15-52:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
[]
val set_goal : fol formula -> goals = <fun>
val extract_thm : goals -> thm = <fun>
val tac_proof : goals -> (goals -> goals) list -> thm = <fun>
val prove : fol formula -> (goals -> goals) list -> thm = <fun>
File "/home/student/Harrison/OCaml/tactics.ml", line 57, characters 11-84:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
[]
File "/home/student/Harrison/OCaml/tactics.ml", line 56, characters 19-177:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
Goals ([], _)
val conj_intro_tac : goals -> goals = <fun>
File "/home/student/Harrison/OCaml/tactics.ml", line 65, characters 20-52:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
[]
val jmodify : ('a list -> 'b) -> ('a -> 'a) -> 'a list -> 'b = <fun>
val gen_right_alpha : string -> string -> thm -> thm = <fun>
File "/home/student/Harrison/OCaml/tactics.ml", line 79, characters 23-264:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
Goals ([], _)
val forall_intro_tac : string -> goals -> goals = <fun>
File "/home/student/Harrison/OCaml/tactics.ml", line 91, characters 6-17:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
(Exists (_, _)|Forall (_, _)|Iff (_, _)|Imp (_, _)|Or (_, _)|And (_, _)|
Atom _|True|False)
val right_exists : string -> term -> fol formula -> thm = <fun>
File "/home/student/Harrison/OCaml/tactics.ml", line 100, characters 23-164:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
Goals ([], _)
val exists_intro_tac : term -> goals -> goals = <fun>
File "/home/student/Harrison/OCaml/tactics.ml", line 108, characters 20-171:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
```

```
Goals ([], _)
val imp_intro_tac : string -> goals -> goals = <fun>
File "/home/student/Harrison/OCaml/tactics.ml", line 116, characters 15-86:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
Goals ([], _)
val assumptate : goals -> thm -> thm = <fun>
val firstassum : ('a * fol formula) list -> thm = <fun>
val using : thm list -> 'a -> goals -> thm list = <fun>
val assumps : ('a * fol formula) list -> ('a * thm) list = <fun>
File "/home/student/Harrison/OCaml/tactics.ml", line 153, characters 14-100:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
Goals ([], _)
val by : string list -> 'a -> goals -> thm list = <fun>
val justify :
  ('a -> fol formula -> goals -> thm list) ->
  'a -> fol formula -> goals -> thm = <fun>
File "/home/student/Harrison/OCaml/tactics.ml", line 172, characters 28-42:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
[]
File "/home/student/Harrison/OCaml/tactics.ml", line 171, characters 17-95:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
Goals ([], _)
val proof : (goals -> goals) list -> fol formula -> goals -> thm list =
  <fun>
val at : 'a -> 'b -> 'c -> 'd list = <fun>
val once : 'a list = []
File "/home/student/Harrison/OCaml/tactics.ml", line 184, characters 23-128:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
Goals ([], _)
val auto_tac :
  ('a -> fol formula -> goals -> thm list) -> 'a -> goals -> goals =
  <fun>
File "/home/student/Harrison/OCaml/tactics.ml", line 192, characters 28-223:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
Goals ([], _)
val lemma_tac :
  string ->
  fol formula ->
  ('a -> fol formula -> goals -> thm list) -> 'a -> goals -> goals =
  <fun>
File "/home/student/Harrison/OCaml/tactics.ml", line 202, characters 6-17:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
(Forall (_, _)|Iff (_, _)|Imp (_, _)|Or (_, _)|And (_, _)|Not _|Atom _|True|
False)
File "/home/student/Harrison/OCaml/tactics.ml", line 201, characters 35-381:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
Goals ([], _)
```

```
val exists_elim_tac :
  string ->
  fol formula ->
  ('a -> fol formula -> goals -> thm list) -> 'a -> goals -> goals =
  <fun>
val ante_disj : thm -> thm -> thm = <fun>
File "/home/student/Harrison/OCaml/tactics.ml", line 227, characters 38-45:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
(Exists (_, _)|Forall (_, _)|Iff (_, _)|Imp (_, _)|And (_, _)|Not _|Atom _|
True|False)
File "/home/student/Harrison/OCaml/tactics.ml", line 228, characters 11-128:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
[]
File "/home/student/Harrison/OCaml/tactics.ml", line 226, characters 33-301:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
Goals ([], _)
val disj_elim_tac :
  string ->
  fol formula ->
  ('a -> fol formula -> goals -> thm list) -> 'a -> goals -> goals =
  <fun>
- : dummy_interactive = START_INTERACTIVE
// tactics.p001
val g0 : goals =
1 subgoal:
---> (forall x. x <= x) /\
     (forall x y z. x <= y /\ y <= z ==> x <= z) /\
     (forall x y. f(x) <= y <=> x <= g(y)) ==>
     (forall x y. x <= y ==> f(x) <= f(y)) /\
     (forall x y. x <= y ==> g(x) <= g(y))

// tactics.p002
val g1 : goals =
1 subgoal:
ant: (forall x. x <= x) /\
     (forall x y z. x <= y /\ y <= z ==> x <= z) /\
     (forall x y. f(x) <= y <=> x <= g(y))
---> (forall x y. x <= y ==> f(x) <= f(y)) /\
     (forall x y. x <= y ==> g(x) <= g(y))

// tactics.p003
val g2 : goals =
2 subgoals starting with
ant: (forall x. x <= x) /\
     (forall x y z. x <= y /\ y <= z ==> x <= z) /\
     (forall x y. f(x) <= y <=> x <= g(y))
---> forall x y. x <= y ==> f(x) <= f(y)

// tactics.p004
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
val g3 : goals = No subgoals
// tactics.p005
- : thm =
|-
(forall x. x <= x) /\
(forall x y z. x <= y /\ y <= z ==> x <= z) /\
(forall x y. f(x) <= y <=> x <= g(y)) ==>
(forall x y. x <= y ==> f(x) <= f(y)) /\
(forall x y. x <= y ==> g(x) <= g(y))
// tactics.p006
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
- : thm =
|-
(forall x. x <= x) /\
(forall x y z. x <= y /\ y <= z ==> x <= z) /\
(forall x y. f(x) <= y <=> x <= g(y)) ==>
(forall x y. x <= y ==> f(x) <= f(y)) /\
(forall x y. x <= y ==> g(x) <= g(y))
- : dummy_interactive = END_INTERACTIVE
val multishunt : int -> thm -> thm = <fun>
File "/home/student/Harrison/OCaml/tactics.ml", line 276, characters 15-264:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
Goals ([], _)
val assume : (string * fol formula) list -> goals -> goals = <fun>
val note :
  string * fol formula ->
  ('a -> fol formula -> goals -> thm list) -> 'a -> goals -> goals =
  <fun>
val have :
  fol formula ->
  ('a -> fol formula -> goals -> thm list) -> 'a -> goals -> goals =
  <fun>
File "/home/student/Harrison/OCaml/tactics.ml", line 287, characters 22-106:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
Goals ([], _)
val so :
  ('a -> ('b -> 'c -> goals -> thm list) -> 'd) ->
  'a -> ('b -> 'c -> goals -> thm list) -> 'd = <fun>
val fix : string -> goals -> goals = <fun>
val consider :
  string * fol formula ->
  ('a -> fol formula -> goals -> thm list) -> 'a -> goals -> goals =
  <fun>
val take : term -> goals -> goals = <fun>
val cases :
  fol formula ->
  ('_a -> fol formula -> goals -> thm list) -> '_a -> goals -> goals =
  <fun>
File "/home/student/Harrison/OCaml/tactics.ml", line 302, characters 25-349:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
Goals ([], _)
val conclude :
  fol formula ->
  ('a -> fol formula -> goals -> thm list) -> 'a -> goals -> goals =
  <fun>
File "/home/student/Harrison/OCaml/tactics.ml", line 314, characters 25-84:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
Goals ([], _)
val our :
  'a ->
  ('b -> fol formula -> goals -> thm list) -> 'b -> goals -> goals =
  <fun>
val thesis : string = ""
File "/home/student/Harrison/OCaml/tactics.ml", line 322, characters 8-154:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
Goals ([], _)
val qed : goals -> goals = <fun>
- : dummy_interactive = START_INTERACTIVE
// tactics.p007
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
val ewd954 : thm =
  |-
  (forall x y. x <= y <=> x * y = x) /\
  (forall x y. f(x * y) = f(x) * f(y)) ==>
  (forall x y. x <= y ==> f(x) <= f(y))
- : dummy_interactive = END_INTERACTIVE
- : dummy_interactive = START_INTERACTIVE
// tactics.p008
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 0
Searching with depth limit 1
- : thm =
|-
(exists x. p(x)) ==>
(forall x. p(x) ==> p(f(x))) ==> (exists y. p(f(f(f(f(y))))))
File "/home/student/Harrison/OCaml/tactics.ml", line 378, characters 17-46:
Warning Y: unused variable gl.
File "/home/student/Harrison/OCaml/tactics.ml", line 380, characters 8-99:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
[]
File "/home/student/Harrison/OCaml/tactics.ml", line 378, characters 16-188:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
Goals ([], _)
// tactics.p009
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 0
Searching with depth limit 1
- : thm =
|-
(exists x. p(x)) ==>
(forall x. p(x) ==> p(f(x))) ==> (exists y. p(f(f(f(f(y))))))
val run : ('a -> 'a) list -> 'a -> 'a = <fun>
val current_goal : goals list ref = {contents = [
1 subgoal:
---> false
]}
val g : fol formula -> goals = <fun>
val e : (goals -> goals) -> goals = <fun>
val es : (goals -> goals) list -> goals = <fun>
val b : unit -> goals = <fun>
// tactics.p011
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
- : thm =
|-
p(a) ==> (forall x. p(x) ==> p(f(x))) ==> (exists y. p(y) /\ p(f(y)))
// tactics.p012
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 0
Searching with depth limit 1
- : thm =
|-
(exists x. p(x)) ==>
(forall x. p(x) ==> p(f(x))) ==> (exists y. p(f(f(f(f(y))))))
// tactics.p013
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 0
- : thm =
|-
forall a.
  p(a) ==> (forall x. p(x) ==> p(f(x))) ==> (exists y. p(y) /\ p(f(y)))
// tactics.p014
Searching with depth limit 0
Searching with depth limit 1
- : thm =
|-
p(c) ==> (forall x. p(x) ==> p(f(x))) ==> (exists y. p(y) /\ p(f(y)))
// tactics.p015
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 0
- : thm =
|-
forall a.
  p(a) ==> (forall x. p(x) ==> p(f(x))) ==> (exists y. p(y) /\ p(f(y)))
// tactics.p016
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 0
- : thm =
|-
forall a.
  p(a) ==> (forall x. p(x) ==> p(f(x))) ==> (exists y. p(y) /\ p(f(y)))
// tactics.p017
- : thm = |- p(a) \/ p(b) ==> q ==> (exists y. p(y))
// tactics.p018
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 0
Searching with depth limit 0
Searching with depth limit 1
- : thm =
|-
(p(a) \/ p(b)) /\ (forall x. p(x) ==> p(f(x))) ==> (exists y. p(f(y)))
// tactics.p019
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 0
Searching with depth limit 1
- : thm =
|-
(exists x. p(x)) ==>
(forall x. p(x) ==> p(f(x))) ==> (exists y. p(f(y)))
// tactics.p020
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 0
- : thm =
|-
(forall x. p(x) ==> q(x)) ==>
(forall x. q(x) ==> p(x)) ==> (p(a) <=> q(a))
- : dummy_interactive = END_INTERACTIVE
#
```

limitations.ml

```
val numeral : num -> term = <fun>
val number : string -> num = <fun>
val pair : num -> num -> num = <fun>
val gterm : term -> num = <fun>
val gform : fol formula -> num = <fun>
- : dummy_interactive = START_INTERACTIVE
// limitations.p001
```

```
- : num = 211657477112832548793799435729494
- : dummy_interactive = END_INTERACTIVE
- : dummy_interactive = START_INTERACTIVE
// limitations.p002
- : num = 735421674029290002
// limitations.p003
- : num = 1767
- : dummy_interactive = END_INTERACTIVE
val gnumeral : num -> num = <fun>
val diag : string -> string = <fun>
- : dummy_interactive = START_INTERACTIVE
// limitations.p004
- : string = "p(`p(x)')"
// limitations.p005
- : string = "This string is diag(`This string is diag(x)')"
- : dummy_interactive = END_INTERACTIVE
val phi : string = "P(diag(`P(diag(x))'))"
val qdiag : string -> string = <fun>
val phi : string = "let `x' be `P(qdiag(x))' in P(qdiag(x))"
- : dummy_interactive = START_INTERACTIVE
// limitations.p006
- : string =
"The result of substituting the quotation of `The result of substituting the quotation of
x for `x' in x has property P' for `x' in `The result of substituting the quotation of x
for `x' in x has property P' has property P"
- : dummy_interactive = END_INTERACTIVE
(fun s -> Printf.printf "%s\n%S\n" s s)
"(fun s -> Printf.printf \"%s\\n%S\\n\" s s)"
- : unit = ()
val diag : string -> fol formula -> fol formula = <fun>
val qdiag : string -> fol formula -> fol formula = <fun>
val dtermval : (string, num) func -> term -> num = <fun>
val dholds : (string, num) func -> fol formula -> bool = <fun>
val dhquant :
  ((num -> bool) -> num list -> bool) ->
  (string, num) func ->
  string -> string -> string -> term -> fol formula -> bool = <fun>
- : dummy_interactive = START_INTERACTIVE
val prime_form : int -> fol formula = <fun>
// limitations.p007
- : bool = false
// limitations.p008
- : bool = true
- : dummy_interactive = END_INTERACTIVE
type formulaclass = Sigma | Pi | Delta
val opp : formulaclass -> formulaclass = <fun>
val classify : formulaclass -> int -> fol formula -> bool = <fun>
- : dummy_interactive = START_INTERACTIVE
// limitations.p009
- : bool = true
- : dummy_interactive = END_INTERACTIVE
File "limitations.ml", line 215, characters 2-936:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
(Exists (_, _)|Forall (_, _))
```

```
(However, some guarded clause may match this value.)
val veref :
  (bool -> bool) -> num -> (string, num) func -> fol formula -> bool =
  <fun>
val verefboundquant :
  num ->
  (string, num) func ->
  string ->
  string -> string -> term -> (bool -> bool) -> fol formula -> bool =
  <fun>
val sholds : num -> (string, num) func -> fol formula -> bool = <fun>
val sigma_bound : fol formula -> num = <fun>
- : dummy_interactive = START_INTERACTIVE
// limitations.p010
- : num = 4
- : dummy_interactive = END_INTERACTIVE
type symbol = Blank | One
type direction = Left | Right | Stay
type tape = Tape of int * (int, symbol) func
val look : tape -> symbol = <fun>
val write : symbol -> tape -> tape = <fun>
val move : direction -> tape -> tape = <fun>
type config = Config of int * tape
val run :
  (int * symbol, symbol * direction * int) func -> config -> config =
  <fun>
val input_tape : int list -> tape = <fun>
val output_tape : tape -> int = <fun>
val exec :
  (int * symbol, symbol * direction * int) func -> int list -> int =
  <fun>
- : dummy_interactive = START_INTERACTIVE
val prog_suc : (int * symbol, symbol * direction * int) func = <func>
// limitations.p011
- : int = 1
// limitations.p012
- : int = 2
// limitations.p013
- : int = 20
- : dummy_interactive = END_INTERACTIVE
val robinson : fol formula =
  <<(forall m n. S(m) = S(n) ==> m = n) /\
    (forall n. ~n = 0 <=> (exists m. n = S(m))) /\
    (forall n. 0 + n = n) /\
    (forall m n. S(m) + n = S(m + n)) /\
    (forall n. 0 * n = 0) /\
    (forall m n. S(m) * n = n + m * n) /\
    (forall m n. m <= n <=> (exists d. m + d = n)) /\
    (forall m n. m < n <=> S(m) <= n)>>
File "limitations.ml", line 379, characters 4-77:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
[]
val suc_inj : thm =
  |-
  (forall m n. S(m) = S(n) ==> m = n) /\
  (forall n. ~n = 0 <=> (exists m. n = S(m))) /\
  (forall n. 0 + n = n) /\
  (forall m n. S(m) + n = S(m + n)) /\
  (forall n. 0 * n = 0) /\
  (forall m n. S(m) * n = n + m * n) /\
  (forall m n. m <= n <=> (exists d. m + d = n)) /\
  (forall m n. m < n <=> S(m) <= n) ==>
  (forall m n. S(m) = S(n) ==> m = n)
val num_cases : thm =
  |-
  (forall m n. S(m) = S(n) ==> m = n) /\
  (forall n. ~n = 0 <=> (exists m. n = S(m))) /\
  (forall n. 0 + n = n) /\
  (forall m n. S(m) + n = S(m + n)) /\
  (forall n. 0 * n = 0) /\
  (forall m n. S(m) * n = n + m * n) /\
  (forall m n. m <= n <=> (exists d. m + d = n)) /\
  (forall m n. m < n <=> S(m) <= n) ==>
  (forall n. ~n = 0 <=> (exists m. n = S(m)))
val add_0 : thm =
  |-
  (forall m n. S(m) = S(n) ==> m = n) /\
  (forall n. ~n = 0 <=> (exists m. n = S(m))) /\
  (forall n. 0 + n = n) /\
  (forall m n. S(m) + n = S(m + n)) /\
  (forall n. 0 * n = 0) /\
  (forall m n. S(m) * n = n + m * n) /\
  (forall m n. m <= n <=> (exists d. m + d = n)) /\
  (forall m n. m < n <=> S(m) <= n) ==> (forall n. 0 + n = n)
val add_suc : thm =
  |-
  (forall m n. S(m) = S(n) ==> m = n) /\
  (forall n. ~n = 0 <=> (exists m. n = S(m))) /\
  (forall n. 0 + n = n) /\
  (forall m n. S(m) + n = S(m + n)) /\
  (forall n. 0 * n = 0) /\
  (forall m n. S(m) * n = n + m * n) /\
  (forall m n. m <= n <=> (exists d. m + d = n)) /\
  (forall m n. m < n <=> S(m) <= n) ==>
  (forall m n. S(m) + n = S(m + n))
val mul_0 : thm =
  |-
  (forall m n. S(m) = S(n) ==> m = n) /\
  (forall n. ~n = 0 <=> (exists m. n = S(m))) /\
  (forall n. 0 + n = n) /\
  (forall m n. S(m) + n = S(m + n)) /\
  (forall n. 0 * n = 0) /\
  (forall m n. S(m) * n = n + m * n) /\
  (forall m n. m <= n <=> (exists d. m + d = n)) /\
  (forall m n. m < n <=> S(m) <= n) ==> (forall n. 0 * n = 0)
val mul_suc : thm =
  |-
  (forall m n. S(m) = S(n) ==> m = n) /\
  (forall n. ~n = 0 <=> (exists m. n = S(m))) /\
  (forall n. 0 + n = n) /\
  (forall m n. S(m) + n = S(m + n)) /\
  (forall n. 0 * n = 0) /\
  (forall m n. S(m) * n = n + m * n) /\
  (forall m n. m <= n <=> (exists d. m + d = n)) /\
  (forall m n. m < n <=> S(m) <= n) ==>
  (forall m n. S(m) * n = n + m * n)
val le_def : thm =
  |-
  (forall m n. S(m) = S(n) ==> m = n) /\
  (forall n. ~n = 0 <=> (exists m. n = S(m))) /\
  (forall n. 0 + n = n) /\
  (forall m n. S(m) + n = S(m + n)) /\
  (forall n. 0 * n = 0) /\
  (forall m n. S(m) * n = n + m * n) /\
  (forall m n. m <= n <=> (exists d. m + d = n)) /\
  (forall m n. m < n <=> S(m) <= n) ==>
  (forall m n. m <= n <=> (exists d. m + d = n))
val lt_def : thm =
  |-
  (forall m n. S(m) = S(n) ==> m = n) /\
  (forall n. ~n = 0 <=> (exists m. n = S(m))) /\
  (forall n. 0 + n = n) /\
  (forall m n. S(m) + n = S(m + n)) /\
  (forall n. 0 * n = 0) /\
  (forall m n. S(m) * n = n + m * n) /\
  (forall m n. m <= n <=> (exists d. m + d = n)) /\
  (forall m n. m < n <=> S(m) <= n) ==>
  (forall m n. m < n <=> S(m) <= n)
val right_spec : term -> thm -> thm = <fun>
val right_mp : thm -> thm -> thm = <fun>
val right_imp_trans : thm -> thm -> thm = <fun>
val right_sym : thm -> thm = <fun>
val right_trans : thm -> thm -> thm = <fun>
File "limitations.ml", line 407, characters 2-347:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
Var _
val robop : term -> thm = <fun>
val robeval : term -> thm = <fun>
- : dummy_interactive = START_INTERACTIVE
// limitations.p014
- : thm =
|-
(forall m n. S(m) = S(n) ==> m = n) /\
(forall n. ~n = 0 <=> (exists m. n = S(m))) /\
(forall n. 0 + n = n) /\
(forall m n. S(m) + n = S(m + n)) /\
(forall n. 0 * n = 0) /\
(forall m n. S(m) * n = n + m * n) /\
(forall m n. m <= n <=> (exists d. m + d = n)) /\
(forall m n. m < n <=> S(m) <= n) ==> S(0) + S(S(0)) *
(S(0) + S(S(0)) + S(0)) = S(S(S(S(S(S(S(S(S(0)))))))))
- : dummy_interactive = END_INTERACTIVE
val robinson_consequences : fol formula =
  <<(forall n. S(n) = 0 ==> false) /\
    (forall n. 0 = S(n) ==> false) /\
    (forall m n. (m = n ==> false) ==> S(m) = S(n) ==> false) /\
    (forall m n. (exists d. m + d = n) ==> m <= n) /\
    (forall m n. S(m) <= n ==> m < n) /\
    (forall m n.
       (forall d. d <= n ==> d = m ==> false) ==> m <= n ==> false) /\
    (forall m n.
       (forall d. d < n ==> d = m ==> false) ==> m < n ==> false) /\
    (forall n. n <= 0 \/ (exists m. S(m) = n)) /\
    (forall n. n <= 0 ==> n = 0) /\
    (forall m n. S(m) <= S(n) ==> m <= n) /\
    (forall m n. m < S(n) ==> m <= n) /\ (forall n. n < 0 ==> false)>>
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
Searching with depth limit 0
Searching with depth limit 1
val robinson_thm : thm =
  |-
  (forall m n. S(m) = S(n) ==> m = n) /\
  (forall n. ~n = 0 <=> (exists m. n = S(m))) /\
  (forall n. 0 + n = n) /\
  (forall m n. S(m) + n = S(m + n)) /\
  (forall n. 0 * n = 0) /\
  (forall m n. S(m) * n = n + m * n) /\
  (forall m n. m <= n <=> (exists d. m + d = n)) /\
  (forall m n. m < n <=> S(m) <= n) ==>
  (forall n. S(n) = 0 ==> false) /\
  (forall n. 0 = S(n) ==> false) /\
  (forall m n. (m = n ==> false) ==> S(m) = S(n) ==> false) /\
  (forall m n. (exists d. m + d = n) ==> m <= n) /\
  (forall m n. S(m) <= n ==> m < n) /\
  (forall m n.
     (forall d. d <= n ==> d = m ==> false) ==> m <= n ==> false) /\
  (forall m n.
     (forall d. d < n ==> d = m ==> false) ==> m < n ==> false) /\
  (forall n. n <= 0 \/ (exists m. S(m) = n)) /\
  (forall n. n <= 0 ==> n = 0) /\
  (forall m n. S(m) <= S(n) ==> m <= n) /\
  (forall m n. m < S(n) ==> m <= n) /\ (forall n. n < 0 ==> false)
File "limitations.ml", line 543, characters 4-134:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
[]
val suc_0_l : thm =
  |-
  (forall m n. S(m) = S(n) ==> m = n) /\
  (forall n. ~n = 0 <=> (exists m. n = S(m))) /\
  (forall n. 0 + n = n) /\
  (forall m n. S(m) + n = S(m + n)) /\
  (forall n. 0 * n = 0) /\
  (forall m n. S(m) * n = n + m * n) /\
  (forall m n. m <= n <=> (exists d. m + d = n)) /\
  (forall m n. m < n <=> S(m) <= n) ==> (forall n. S(n) = 0 ==> false)
val suc_0_r : thm =
  |-
  (forall m n. S(m) = S(n) ==> m = n) /\
  (forall n. ~n = 0 <=> (exists m. n = S(m))) /\
  (forall n. 0 + n = n) /\
  (forall m n. S(m) + n = S(m + n)) /\
  (forall n. 0 * n = 0) /\
  (forall m n. S(m) * n = n + m * n) /\
  (forall m n. m <= n <=> (exists d. m + d = n)) /\
  (forall m n. m < n <=> S(m) <= n) ==> (forall n. 0 = S(n) ==> false)
val suc_inj_false : thm =
  |-
  (forall m n. S(m) = S(n) ==> m = n) /\
  (forall n. ~n = 0 <=> (exists m. n = S(m))) /\
  (forall n. 0 + n = n) /\
  (forall m n. S(m) + n = S(m + n)) /\
  (forall n. 0 * n = 0) /\
  (forall m n. S(m) * n = n + m * n) /\
  (forall m n. m <= n <=> (exists d. m + d = n)) /\
  (forall m n. m < n <=> S(m) <= n) ==>
  (forall m n. (m = n ==> false) ==> S(m) = S(n) ==> false)
val expand_le : thm =
  |-
  (forall m n. S(m) = S(n) ==> m = n) /\
  (forall n. ~n = 0 <=> (exists m. n = S(m))) /\
  (forall n. 0 + n = n) /\
  (forall m n. S(m) + n = S(m + n)) /\
  (forall n. 0 * n = 0) /\
  (forall m n. S(m) * n = n + m * n) /\
  (forall m n. m <= n <=> (exists d. m + d = n)) /\
  (forall m n. m < n <=> S(m) <= n) ==>
  (forall m n. (exists d. m + d = n) ==> m <= n)
val expand_lt : thm =
  |-
  (forall m n. S(m) = S(n) ==> m = n) /\
  (forall n. ~n = 0 <=> (exists m. n = S(m))) /\
  (forall n. 0 + n = n) /\
  (forall m n. S(m) + n = S(m + n)) /\
  (forall n. 0 * n = 0) /\
  (forall m n. S(m) * n = n + m * n) /\
  (forall m n. m <= n <=> (exists d. m + d = n)) /\
  (forall m n. m < n <=> S(m) <= n) ==>
  (forall m n. S(m) <= n ==> m < n)
val expand_nle : thm =
  |-
  (forall m n. S(m) = S(n) ==> m = n) /\
  (forall n. ~n = 0 <=> (exists m. n = S(m))) /\
  (forall n. 0 + n = n) /\
  (forall m n. S(m) + n = S(m + n)) /\
  (forall n. 0 * n = 0) /\
  (forall m n. S(m) * n = n + m * n) /\
  (forall m n. m <= n <=> (exists d. m + d = n)) /\
  (forall m n. m < n <=> S(m) <= n) ==>
  (forall m n.
     (forall d. d <= n ==> d = m ==> false) ==> m <= n ==> false)
val expand_nlt : thm =
  |-
  (forall m n. S(m) = S(n) ==> m = n) /\
  (forall n. ~n = 0 <=> (exists m. n = S(m))) /\
  (forall n. 0 + n = n) /\
  (forall m n. S(m) + n = S(m + n)) /\
  (forall n. 0 * n = 0) /\
  (forall m n. S(m) * n = n + m * n) /\
  (forall m n. m <= n <=> (exists d. m + d = n)) /\
  (forall m n. m < n <=> S(m) <= n) ==>
  (forall m n.
     (forall d. d < n ==> d = m ==> false) ==> m < n ==> false)
val num_lecases : thm =
  |-
  (forall m n. S(m) = S(n) ==> m = n) /\
  (forall n. ~n = 0 <=> (exists m. n = S(m))) /\
  (forall n. 0 + n = n) /\
  (forall m n. S(m) + n = S(m + n)) /\
  (forall n. 0 * n = 0) /\
  (forall m n. S(m) * n = n + m * n) /\
  (forall m n. m <= n <=> (exists d. m + d = n)) /\
  (forall m n. m < n <=> S(m) <= n) ==>
  (forall n. n <= 0 \/ (exists m. S(m) = n))
val le_0 : thm =
  |-
  (forall m n. S(m) = S(n) ==> m = n) /\
  (forall n. ~n = 0 <=> (exists m. n = S(m))) /\
  (forall n. 0 + n = n) /\
  (forall m n. S(m) + n = S(m + n)) /\
  (forall n. 0 * n = 0) /\
  (forall m n. S(m) * n = n + m * n) /\
  (forall m n. m <= n <=> (exists d. m + d = n)) /\
  (forall m n. m < n <=> S(m) <= n) ==> (forall n. n <= 0 ==> n = 0)
val le_suc : thm =
  |-
  (forall m n. S(m) = S(n) ==> m = n) /\
  (forall n. ~n = 0 <=> (exists m. n = S(m))) /\
  (forall n. 0 + n = n) /\
  (forall m n. S(m) + n = S(m + n)) /\
  (forall n. 0 * n = 0) /\
  (forall m n. S(m) * n = n + m * n) /\
  (forall m n. m <= n <=> (exists d. m + d = n)) /\
  (forall m n. m < n <=> S(m) <= n) ==>
  (forall m n. S(m) <= S(n) ==> m <= n)
val lt_suc : thm =
  |-
  (forall m n. S(m) = S(n) ==> m = n) /\
  (forall n. ~n = 0 <=> (exists m. n = S(m))) /\
  (forall n. 0 + n = n) /\
  (forall m n. S(m) + n = S(m + n)) /\
  (forall n. 0 * n = 0) /\
  (forall m n. S(m) * n = n + m * n) /\
  (forall m n. m <= n <=> (exists d. m + d = n)) /\
  (forall m n. m < n <=> S(m) <= n) ==>
  (forall m n. m < S(n) ==> m <= n)
val lt_0 : thm =
  |-
  (forall m n. S(m) = S(n) ==> m = n) /\
  (forall n. ~n = 0 <=> (exists m. n = S(m))) /\
  (forall n. 0 + n = n) /\
  (forall m n. S(m) + n = S(m + n)) /\
  (forall n. 0 * n = 0) /\
  (forall m n. S(m) * n = n + m * n) /\
  (forall m n. m <= n <=> (exists d. m + d = n)) /\
  (forall m n. m < n <=> S(m) <= n) ==> (forall n. n < 0 ==> false)
val rob_eq : term -> term -> thm = <fun>
val rob_nen : term * term -> thm = <fun>
val rob_ne : term -> term -> thm = <fun>
- : dummy_interactive = START_INTERACTIVE
// limitations.p015
- : thm =
|-
(forall m n. S(m) = S(n) ==> m = n) /\
(forall n. ~n = 0 <=> (exists m. n = S(m))) /\
(forall n. 0 + n = n) /\
(forall m n. S(m) + n = S(m + n)) /\
(forall n. 0 * n = 0) /\
(forall m n. S(m) * n = n + m * n) /\
(forall m n. m <= n <=> (exists d. m + d = n)) /\
(forall m n. m < n <=> S(m) <= n) ==>
S(0) + S(0) + S(0) = S(S(0)) * S(S(0)) ==> false
// limitations.p016
- : thm =
|-
(forall m n. S(m) = S(n) ==> m = n) /\
(forall n. ~n = 0 <=> (exists m. n = S(m))) /\
(forall n. 0 + n = n) /\
(forall m n. S(m) + n = S(m + n)) /\
(forall n. 0 * n = 0) /\
(forall m n. S(m) * n = n + m * n) /\
(forall m n. m <= n <=> (exists d. m + d = n)) /\
(forall m n. m < n <=> S(m) <= n) ==>
0 + 0 * S(0) = S(S(0)) + 0 ==> false
// limitations.p017
- : thm =
|-
(forall m n. S(m) = S(n) ==> m = n) /\
(forall n. ~n = 0 <=> (exists m. n = S(m))) /\
(forall n. 0 + n = n) /\
(forall m n. S(m) + n = S(m + n)) /\
(forall n. 0 * n = 0) /\
(forall m n. S(m) * n = n + m * n) /\
(forall m n. m <= n <=> (exists d. m + d = n)) /\
(forall m n. m < n <=> S(m) <= n) ==>
S(S(0)) + 0 = 0 + 0 + 0 * 0 ==> false
- : dummy_interactive = END_INTERACTIVE
val introduce_connective : fol formula -> thm = <fun>
val elim_bex : fol formula -> thm = <fun>
val sigma_elim : fol formula -> thm = <fun>
File "limitations.ml", line 622, characters 2-1007:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
((Exists (_, _)|Forall (_, _)|Iff (_, _)|Or (_, _)|And (_, _)|Not _|Atom _|
 True|False),
_)
val boundquant_step : thm -> thm -> thm = <fun>
File "limitations.ml", line 689, characters 2-971:
Warning P: this pattern-matching is not exhaustive.
Here is an example of a value that is not matched:
("", _)
val sigma_prove : fol formula -> thm = <fun>
val bounded_prove : string * string * term * fol formula -> thm = <fun>
val boundednum_prove : string * string * term * fol formula -> thm =
  <fun>
- : dummy_interactive = START_INTERACTIVE
// limitations.p018
- : thm =
|-
(forall m n. S(m) = S(n) ==> m = n) /\
(forall n. ~n = 0 <=> (exists m. n = S(m))) /\
(forall n. 0 + n = n) /\
(forall m n. S(m) + n = S(m + n)) /\
(forall n. 0 * n = 0) /\
(forall m n. S(m) * n = n + m * n) /\
(forall m n. m <= n <=> (exists d. m + d = n)) /\
(forall m n. m < n <=> S(m) <= n) ==>
(exists p.
   S(S(0)) <= p /\
   (forall n. n < p ==> (exists x. x <= p /\ p = n * x) ==> n = S(0)))
- : dummy_interactive = END_INTERACTIVE
- : dummy_interactive = START_INTERACTIVE
// limitations.p019
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
- : int list = [5; 5]
- : dummy_interactive = END_INTERACTIVE
- : dummy_interactive = START_INTERACTIVE
// limitations.p020
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
Searching with depth limit 8
Searching with depth limit 9
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
Searching with depth limit 8
Searching with depth limit 9
Searching with depth limit 10
Searching with depth limit 11
Searching with depth limit 12
Searching with depth limit 13
Searching with depth limit 14
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
Searching with depth limit 8
Searching with depth limit 9
Searching with depth limit 10
Searching with depth limit 11
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
Searching with depth limit 8
Searching with depth limit 9
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
Searching with depth limit 8
Searching with depth limit 0
Searching with depth limit 1
Searching with depth limit 2
Searching with depth limit 3
Searching with depth limit 4
Searching with depth limit 5
Searching with depth limit 6
Searching with depth limit 7
val godel_2 : thm =
  |-
  (forall p. |--(p) ==> |--(Pr(p))) /\
  (forall p q. |--(imp(Pr(imp(p,q)),imp(Pr(p),Pr(q))))) /\
  (forall p. |--(imp(Pr(p),Pr(Pr(p))))) ==>
  (forall p q. |--(imp(p,q)) /\ |--(p) ==> |--(q)) /\
  (forall p q. |--(imp(q,imp(p,q)))) /\
  (forall p q r. |--(imp(imp(p,imp(q,r)),imp(imp(p,q),imp(p,r))))) ==>
  |--(imp(G,imp(Pr(G),F))) /\ |--(imp(imp(Pr(G),F),G)) ==>
  |--(imp(Pr(F),F)) ==> |--(F)
- : dummy_interactive = END_INTERACTIVE
```