LCL axioms

LCL001-0.ax Wajsberg algebra
$\text{implies}(\text{truth}, x) = x \qquad \text{cnf}(\text{wajsberg}_1, \text{axiom})$
$\text{implies}(\text{implies}(x, y), \text{implies}(\text{implies}(y, z), \text{implies}(x, z))) = \text{truth} \qquad \text{cnf}(\text{wajsberg}_2, \text{axiom})$
$\text{implies}(\text{implies}(x, y), y) = \text{implies}(\text{implies}(y, x), x) \qquad \text{cnf}(\text{wajsberg}_3, \text{axiom})$
$\text{implies}(\text{implies}(\text{not}(x), \text{not}(y)), \text{implies}(y, x)) = \text{truth} \qquad \text{cnf}(\text{wajsberg}_4, \text{axiom})$

LCL001-1.ax Wajsberg algebra lattice structure definitions
$\text{big\_V}(x, y) = \text{implies}(\text{implies}(x, y), y) \qquad \text{cnf}(\text{big\_V\_definition}, \text{axiom})$
$\text{big\_hat}(x, y) = \text{not}(\text{big\_V}(\text{not}(x), \text{not}(y))) \qquad \text{cnf}(\text{big\_hat\_definition}, \text{axiom})$
$\text{ordered}(x, y) \Rightarrow \text{implies}(x, y) = \text{truth} \qquad \text{cnf}(\text{partial\_order\_definition}_1, \text{axiom})$
$\text{implies}(x, y) = \text{truth} \Rightarrow \text{ordered}(x, y) \qquad \text{cnf}(\text{partial\_order\_definition}_2, \text{axiom})$

LCL001-2.ax Wajsberg algebra AND and OR definitions
$\text{or}(x, y) = \text{implies}(\text{not}(x), y) \qquad \text{cnf}(\text{or\_definition}, \text{axiom})$
$\text{or}(\text{or}(x, y), z) = \text{or}(x, \text{or}(y, z)) \qquad \text{cnf}(\text{or\_associativity}, \text{axiom})$
$\text{or}(x, y) = \text{or}(y, x) \qquad \text{cnf}(\text{or\_commutativity}, \text{axiom})$
$\text{and}(x, y) = \text{not}(\text{or}(\text{not}(x), \text{not}(y))) \qquad \text{cnf}(\text{and\_definition}, \text{axiom})$
$\text{and}(\text{and}(x, y), z) = \text{and}(x, \text{and}(y, z)) \qquad \text{cnf}(\text{and\_associativity}, \text{axiom})$
$\text{and}(x, y) = \text{and}(y, x) \qquad \text{cnf}(\text{and\_commutativity}, \text{axiom})$

LCL002-0.ax Alternative Wajsberg algebra
$\text{not}(x) = \text{xor}(x, \text{truth}) \qquad \text{cnf}(\text{axiom}_1, \text{axiom})$
$\text{xor}(x, \text{falsehood}) = x \qquad \text{cnf}(\text{axiom}_2, \text{axiom})$
$\text{xor}(x, x) = \text{falsehood} \qquad \text{cnf}(\text{axiom}_3, \text{axiom})$
$\text{and\_star}(x, \text{truth}) = x \qquad \text{cnf}(\text{axiom}_4, \text{axiom})$
$\text{and\_star}(x, \text{falsehood}) = \text{falsehood} \qquad \text{cnf}(\text{axiom}_5, \text{axiom})$
$\text{and\_star}(\text{xor}(\text{truth}, x), x) = \text{falsehood} \qquad \text{cnf}(\text{axiom}_6, \text{axiom})$
$\text{xor}(x, \text{xor}(\text{truth}, y)) = \text{xor}(\text{xor}(x, \text{truth}), y) \qquad \text{cnf}(\text{axiom}_7, \text{axiom})$
$\text{and\_star}(\text{xor}(\text{and\_star}(\text{xor}(\text{truth}, x), y), \text{truth}), y) = \text{and\_star}(\text{xor}(\text{and\_star}(\text{xor}(\text{truth}, y), x), \text{truth}), x) \qquad \text{cnf}(\text{axiom}_8, \text{axiom})$

LCL002-1.ax Alternative Wajsberg algebra definitions
$\text{xor}(x, y) = \text{or}(\text{and}(x, \text{not}(y)), \text{and}(\text{not}(x), y)) \qquad \text{cnf}(\text{xor\_definition}, \text{axiom})$
$\text{xor}(x, y) = \text{xor}(y, x) \qquad \text{cnf}(\text{xor\_commutativity}, \text{axiom})$
$\text{and\_star}(x, y) = \text{not}(\text{or}(\text{not}(x), \text{not}(y))) \qquad \text{cnf}(\text{and\_star\_definition}, \text{axiom})$
$\text{and\_star}(\text{and\_star}(x, y), z) = \text{and\_star}(x, \text{and\_star}(y, z)) \qquad \text{cnf}(\text{and\_star\_associativity}, \text{axiom})$
$\text{and\_star}(x, y) = \text{and\_star}(y, x) \qquad \text{cnf}(\text{and\_star\_commutativity}, \text{axiom})$
$\text{not}(\text{truth}) = \text{falsehood} \qquad \text{cnf}(\text{false\_definition}, \text{axiom})$

LCL003-0.ax Propositional logic deduction
$\text{axiom}(\text{or}(\text{not}(\text{or}(a, a)), a)) \qquad \text{cnf}(\text{axiom\_1}_2, \text{axiom})$
$\text{axiom}(\text{or}(\text{not}(a), \text{or}(b, a))) \qquad \text{cnf}(\text{axiom\_1}_3, \text{axiom})$
$\text{axiom}(\text{or}(\text{not}(\text{or}(a, b)), \text{or}(b, a))) \qquad \text{cnf}(\text{axiom\_1}_4, \text{axiom})$
$\text{axiom}(\text{or}(\text{not}(\text{or}(a, \text{or}(b, c))), \text{or}(b, \text{or}(a, c)))) \qquad \text{cnf}(\text{axiom\_1}_5, \text{axiom})$
$\text{axiom}(\text{or}(\text{not}(\text{or}(\text{not}(a), b)), \text{or}(\text{not}(\text{or}(c, a)), \text{or}(c, b)))) \qquad \text{cnf}(\text{axiom\_1}_6, \text{axiom})$
$\text{axiom}(x) \Rightarrow \text{theorem}(x) \qquad \text{cnf}(\text{rule}_1, \text{axiom})$
$(\text{axiom}(\text{or}(\text{not}(y), x)) \text{ and } \text{theorem}(y)) \Rightarrow \text{theorem}(x) \qquad \text{cnf}(\text{rule}_2, \text{axiom})$
$(\text{axiom}(\text{or}(\text{not}(x), y)) \text{ and } \text{theorem}(\text{or}(\text{not}(y), z))) \Rightarrow \text{theorem}(\text{or}(\text{not}(x), z)) \qquad \text{cnf}(\text{rule}_3, \text{axiom})$

LCL004-0.ax Propositional logic deduction
$\text{axiom}(\text{implies}(\text{or}(a, a), a)) \qquad \text{cnf}(\text{axiom\_1}_2, \text{axiom})$
$\text{axiom}(\text{implies}(a, \text{or}(b, a))) \qquad \text{cnf}(\text{axiom\_1}_3, \text{axiom})$
$\text{axiom}(\text{implies}(\text{or}(a, b), \text{or}(b, a))) \qquad \text{cnf}(\text{axiom\_1}_4, \text{axiom})$
$\text{axiom}(\text{implies}(\text{or}(a, \text{or}(b, c)), \text{or}(b, \text{or}(a, c)))) \qquad \text{cnf}(\text{axiom\_1}_5, \text{axiom})$
$\text{axiom}(\text{implies}(\text{implies}(a, b), \text{implies}(\text{or}(c, a), \text{or}(c, b)))) \qquad \text{cnf}(\text{axiom\_1}_6, \text{axiom})$
$\text{implies}(x, y) = \text{or}(\text{not}(x), y) \qquad \text{cnf}(\text{implies\_definition}, \text{axiom})$
$\text{axiom}(x) \Rightarrow \text{theorem}(x) \qquad \text{cnf}(\text{rule}_1, \text{axiom})$
$(\text{theorem}(\text{implies}(y, x)) \text{ and } \text{theorem}(y)) \Rightarrow \text{theorem}(x) \qquad \text{cnf}(\text{rule}_2, \text{axiom})$

LCL004-1.ax Propositional logic deduction axioms for AND
$\text{and}(p, q) = \text{not}(\text{or}(\text{not}(p), \text{not}(q))) \qquad \text{cnf}(\text{and\_defn}, \text{axiom})$

LCL004-2.ax Propositional logic deduction axioms for EQUIVALENT

equivalent(p, q) = and(implies(p, q), implies(q, p))   cnf(equivalent_defn, axiom)

LCL005-0.ax Propositional logic
c_PropLog_Opl_Ofalse $\neq$ c_PropLog_Opl_Ovar(v_a_H, t_a)   cnf(cls_PropLog_Opl_Odistinct__1__iff1$_0$, axiom)
c_PropLog_Opl_Ovar(v_a_H, t_a) $\neq$ c_PropLog_Opl_Ofalse   cnf(cls_PropLog_Opl_Odistinct__2__iff1$_0$, axiom)
c_PropLog_Opl_Ofalse $\neq$ c_PropLog_Opl_Oop_A_N$_{62}$(v_pl1_H, v_pl2_H, t_a)   cnf(cls_PropLog_Opl_Odistinct__3__iff1$_0$, axiom
c_PropLog_Opl_Oop_A_N$_{62}$(v_pl1_H, v_pl2_H, t_a) $\neq$ c_PropLog_Opl_Ofalse   cnf(cls_PropLog_Opl_Odistinct__4__iff1$_0$, axiom
c_PropLog_Opl_Ovar(v_a, t_a) $\neq$ c_PropLog_Opl_Oop_A_N$_{62}$(v_pl1_H, v_pl2_H, t_a)   cnf(cls_PropLog_Opl_Odistinct__5__iff1
c_PropLog_Opl_Oop_A_N$_{62}$(v_pl1_H, v_pl2_H, t_a) $\neq$ c_PropLog_Opl_Ovar(v_a, t_a)   cnf(cls_PropLog_Opl_Odistinct__6__iff1
c_PropLog_Opl_Ovar(v_a, t_a) = c_PropLog_Opl_Ovar(v_a_H, t_a) $\Rightarrow$ v_a = v_a_H   cnf(cls_PropLog_Opl_Oinject__1__iff1$_0$,
c_PropLog_Opl_Oop_A_N$_{62}$(v_pl$_1$, v_pl$_2$, t_a) = c_PropLog_Opl_Oop_A_N$_{62}$(v_pl1_H, v_pl2_H, t_a) $\Rightarrow$ v_pl$_1$ = v_pl1_H   cnf(
c_PropLog_Opl_Oop_A_N$_{62}$(v_pl$_1$, v_pl$_2$, t_a) = c_PropLog_Opl_Oop_A_N$_{62}$(v_pl1_H, v_pl2_H, t_a) $\Rightarrow$ v_pl$_2$ = v_pl2_H   cnf(
c_in(v_p, v_H, tc_PropLog_Opl(t_a)) $\Rightarrow$ c_in(v_p, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a))   cnf(cls_PropLog_Oth

LCL006+0.ax Propositional logic rules and axioms
modus_ponens $\iff \forall x, y$: ((is_a_theorem(x) and is_a_theorem(implies(x, y))) $\Rightarrow$ is_a_theorem(y))   fof(modus_ponens, axi
substitution_of_equivalents $\iff \forall x, y$: (is_a_theorem(equiv(x, y)) $\Rightarrow x = y$)   fof(substitution_of_equivalents, axiom)
modus_tollens $\iff \forall x, y$: is_a_theorem(implies(implies(not(y), not(x)), implies(x, y)))   fof(modus_tollens, axiom)
implies$_1$ $\iff \forall x, y$: is_a_theorem(implies(x, implies(y, x)))   fof(implies$_1$, axiom)
implies$_2$ $\iff \forall x, y$: is_a_theorem(implies(implies(x, implies(x, y)), implies(x, y)))   fof(implies$_2$, axiom)
implies$_3$ $\iff \forall x, y, z$: is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z))))   fof(implies$_3$, axiom)
and$_1$ $\iff \forall x, y$: is_a_theorem(implies(and(x, y), x))   fof(and$_1$, axiom)
and$_2$ $\iff \forall x, y$: is_a_theorem(implies(and(x, y), y))   fof(and$_2$, axiom)
and$_3$ $\iff \forall x, y$: is_a_theorem(implies(x, implies(y, and(x, y))))   fof(and$_3$, axiom)
or$_1$ $\iff \forall x, y$: is_a_theorem(implies(x, or(x, y)))   fof(or$_1$, axiom)
or$_2$ $\iff \forall x, y$: is_a_theorem(implies(y, or(x, y)))   fof(or$_2$, axiom)
or$_3$ $\iff \forall x, y, z$: is_a_theorem(implies(implies(x, z), implies(implies(y, z), implies(or(x, y), z))))   fof(or$_3$, axiom)
equivalence$_1$ $\iff \forall x, y$: is_a_theorem(implies(equiv(x, y), implies(x, y)))   fof(equivalence$_1$, axiom)
equivalence$_2$ $\iff \forall x, y$: is_a_theorem(implies(equiv(x, y), implies(y, x)))   fof(equivalence$_2$, axiom)
equivalence$_3$ $\iff \forall x, y$: is_a_theorem(implies(implies(x, y), implies(implies(y, x), equiv(x, y))))   fof(equivalence$_3$, axiom)
kn$_1$ $\iff \forall p$: is_a_theorem(implies(p, and(p, p)))   fof(kn$_1$, axiom)
kn$_2$ $\iff \forall p, q$: is_a_theorem(implies(and(p, q), p))   fof(kn$_2$, axiom)
kn$_3$ $\iff \forall p, q, r$: is_a_theorem(implies(implies(p, q), implies(not(and(q, r)), not(and(r, p)))))   fof(kn$_3$, axiom)
cn$_1$ $\iff \forall p, q, r$: is_a_theorem(implies(implies(p, q), implies(implies(q, r), implies(p, r))))   fof(cn$_1$, axiom)
cn$_2$ $\iff \forall p, q$: is_a_theorem(implies(p, implies(not(p), q)))   fof(cn$_2$, axiom)
cn$_3$ $\iff \forall p$: is_a_theorem(implies(implies(not(p), p), p))   fof(cn$_3$, axiom)
$r_1 \iff \forall p$: is_a_theorem(implies(or(p, p), p))   fof(r_1, axiom)
$r_2 \iff \forall p, q$: is_a_theorem(implies(q, or(p, q)))   fof(r_2, axiom)
$r_3 \iff \forall p, q$: is_a_theorem(implies(or(p, q), or(q, p)))   fof(r_3, axiom)
$r_4 \iff \forall p, q, r$: is_a_theorem(implies(or(p, or(q, r)), or(q, or(p, r))))   fof(r_4, axiom)
$r_5 \iff \forall p, q, r$: is_a_theorem(implies(implies(q, r), implies(or(p, q), or(p, r))))   fof(r_5, axiom)

LCL006+1.ax Propositional logic definitions
op_or $\Rightarrow \forall x, y$: or(x, y) = not(and(not(x), not(y)))   fof(op_or, axiom)
op_and $\Rightarrow \forall x, y$: and(x, y) = not(or(not(x), not(y)))   fof(op_and, axiom)
op_implies_and $\Rightarrow \forall x, y$: implies(x, y) = not(and(x, not(y)))   fof(op_implies_and, axiom)
op_implies_or $\Rightarrow \forall x, y$: implies(x, y) = or(not(x), y)   fof(op_implies_or, axiom)
op_equiv $\Rightarrow \forall x, y$: equiv(x, y) = and(implies(x, y), implies(y, x))   fof(op_equiv, axiom)

LCL006+2.ax Hilbert's axiomatization of propositional logic
op_or   fof(hilbert_op_or, axiom)
op_implies_and   fof(hilbert_op_implies_and, axiom)
op_equiv   fof(hilbert_op_equiv, axiom)
modus_ponens   fof(hilbert_modus_ponens, axiom)
modus_tollens   fof(hilbert_modus_tollens, axiom)
implies$_1$ fof(hilbert_implies$_1$, axiom)
implies$_2$ fof(hilbert_implies$_2$, axiom)
implies$_3$ fof(hilbert_implies$_3$, axiom)
and$_1$ fof(hilbert_and$_1$, axiom)
and$_2$ fof(hilbert_and$_2$, axiom)
and$_3$ fof(hilbert_and$_3$, axiom)

or$_1$ fof(hilbert_or$_1$, axiom)
or$_2$ fof(hilbert_or$_2$, axiom)
or$_3$ fof(hilbert_or$_3$, axiom)
equivalence$_1$ fof(hilbert_equivalence$_1$, axiom)
equivalence$_2$ fof(hilbert_equivalence$_2$, axiom)
equivalence$_3$ fof(hilbert_equivalence$_3$, axiom)
substitution_of_equivalents fof(substitution_of_equivalents, axiom)

LCL006+3.ax Lukasiewicz's axiomatization of propositional logic
op_or fof(luka_op_or, axiom)
op_implies fof(luka_op_implies, axiom)
op_equiv fof(luka_op_equiv, axiom)
modus_ponens fof(luka_modus_ponens, axiom)
cn$_1$ fof(luka_cn$_1$, axiom)
cn$_2$ fof(luka_cn$_2$, axiom)
cn$_3$ fof(luka_cn$_3$, axiom)
substitution_of_equivalents fof(substitution_of_equivalents, axiom)

LCL006+4.ax Principia's axiomatization of propositional logic
op_implies_or fof(principia_op_implies_or, axiom)
op_and fof(principia_op_and, axiom)
op_equiv fof(principia_op_equiv, axiom)
modus_ponens fof(principia_modus_ponens, axiom)
r_1 fof(principia_r$_1$, axiom)
r_2 fof(principia_r$_2$, axiom)
r_3 fof(principia_r$_3$, axiom)
r_4 fof(principia_r$_4$, axiom)
r_5 fof(principia_r$_5$, axiom)
substitution_of_equivalents fof(substitution_of_equivalents, axiom)

LCL006+5.ax Rosser's axiomatization of propositional logic
op_or fof(rosser_op_or, axiom)
op_implies_and fof(rosser_op_implies_and, axiom)
op_equiv fof(rosser_op_equiv, axiom)
modus_ponens fof(rosser_modus_ponens, axiom)
kn$_1$ fof(rosser_kn$_1$, axiom)
kn$_2$ fof(rosser_kn$_2$, axiom)
kn$_3$ fof(rosser_kn$_3$, axiom)
substitution_of_equivalents fof(substitution_of_equivalents, axiom)

LCL007+0.ax Propositional modal logic rules and axioms
necessitation $\iff \forall x$: (is_a_theorem(x) $\Rightarrow$ is_a_theorem(necessarily(x))) fof(necessitation, axiom)
modus_ponens_strict_implies $\iff \forall x, y$: ((is_a_theorem(x) and is_a_theorem(strict_implies(x, y))) $\Rightarrow$ is_a_theorem(y)) fo
adjunction $\iff \forall x, y$: ((is_a_theorem(x) and is_a_theorem(y)) $\Rightarrow$ is_a_theorem(and(x, y))) fof(adjunction, axiom)
substitution_strict_equiv $\iff \forall x, y$: (is_a_theorem(strict_equiv(x, y)) $\Rightarrow$ $x = y$) fof(substitution_strict_equiv, axiom)
axiom_K $\iff \forall x, y$: is_a_theorem(implies(necessarily(implies(x, y)), implies(necessarily(x), necessarily(y)))) fof(axiom_K
axiom_M $\iff \forall x$: is_a_theorem(implies(necessarily(x), x)) fof(axiom_M, axiom)
axiom$_4$ $\iff \forall x$: is_a_theorem(implies(necessarily(x), necessarily(necessarily(x)))) fof(axiom$_4$, axiom)
axiom_B $\iff \forall x$: is_a_theorem(implies(x, necessarily(possibly(x)))) fof(axiom_B, axiom)
axiom$_5$ $\iff \forall x$: is_a_theorem(implies(possibly(x), necessarily(possibly(x)))) fof(axiom$_5$, axiom)
axiom_s$_1$ $\iff \forall x, y, z$: is_a_theorem(implies(and(necessarily(implies(x, y)), necessarily(implies(y, z))), necessarily(implies(x,
axiom_s$_2$ $\iff \forall p, q$: is_a_theorem(strict_implies(possibly(and(p, q)), and(possibly(p), possibly(q)))) fof(axiom_s$_2$, axiom)
axiom_s$_3$ $\iff \forall x, y$: is_a_theorem(strict_implies(strict_implies(x, y), strict_implies(not(possibly(y)), not(possibly(x))))) fo
axiom_s$_4$ $\iff \forall x$: is_a_theorem(strict_implies(necessarily(x), necessarily(necessarily(x)))) fof(axiom_s$_4$, axiom)
axiom_m$_1$ $\iff \forall x, y$: is_a_theorem(strict_implies(and(x, y), and(y, x))) fof(axiom_m$_1$, axiom)
axiom_m$_2$ $\iff \forall x, y$: is_a_theorem(strict_implies(and(x, y), x)) fof(axiom_m$_2$, axiom)
axiom_m$_3$ $\iff \forall x, y, z$: is_a_theorem(strict_implies(and(and(x, y), z), and(x, and(y, z)))) fof(axiom_m$_3$, axiom)
axiom_m$_4$ $\iff \forall x$: is_a_theorem(strict_implies(x, and(x, x))) fof(axiom_m$_4$, axiom)
axiom_m$_5$ $\iff \forall x, y, z$: is_a_theorem(strict_implies(and(strict_implies(x, y), strict_implies(y, z)), strict_implies(x, z))) fof(
axiom_m$_6$ $\iff \forall x$: is_a_theorem(strict_implies(x, possibly(x))) fof(axiom_m$_6$, axiom)
axiom_m$_7$ $\iff \forall p, q$: is_a_theorem(strict_implies(possibly(and(p, q)), p)) fof(axiom_m$_7$, axiom)

axiom_m$_8$ $\iff$ $\forall p, q$: is_a_theorem(strict_implies(strict_implies(p, q), strict_implies(possibly(p), possibly(q)))) fof(axiom_m
axiom_m$_9$ $\iff$ $\forall x$: is_a_theorem(strict_implies(possibly(possibly(x)), possibly(x))) fof(axiom_m$_9$, axiom)
axiom_m$_{10}$ $\iff$ $\forall x$: is_a_theorem(strict_implies(possibly(x), necessarily(possibly(x)))) fof(axiom_m$_{10}$, axiom)

LCL007+1.ax Propositional modal logic definitions
op_possibly $\Rightarrow$ $\forall x$: possibly(x) = not(necessarily(not(x))) fof(op_possibly, axiom)
op_necessarily $\Rightarrow$ $\forall x$: necessarily(x) = not(possibly(not(x))) fof(op_necessarily, axiom)
op_strict_implies $\Rightarrow$ $\forall x, y$: strict_implies(x, y) = necessarily(implies(x, y)) fof(op_strict_implies, axiom)
op_strict_equiv $\Rightarrow$ $\forall x, y$: strict_equiv(x, y) = and(strict_implies(x, y), strict_implies(y, x)) fof(op_strict_equiv, axiom)

LCL007+2.ax KM5 axiomatization of S5 based on Hilbert's PC
op_possibly fof(km5_op_possibly, axiom)
necessitation fof(km5_necessitation, axiom)
axiom_K fof(km5_axiom_K, axiom)
axiom_M fof(km5_axiom_M, axiom)
axiom$_5$ fof(km5_axiom$_5$, axiom)

LCL007+3.ax KM4B axiomatization of S5 based on Hilbert's PC
op_possibly fof(km4b_op_possibly, axiom)
necessitation fof(km4b_necessitation, axiom)
axiom_K fof(km4b_axiom_K, axiom)
axiom_M fof(km4b_axiom_M, axiom)
axiom$_4$ fof(km4b_axiom$_4$, axiom)
axiom_B fof(km4b_axiom_B, axiom)

LCL007+4.ax Axiomatization of S1-0
op_possibly fof(s1_0_op_possibly, axiom)
op_or fof(s1_0_op_or, axiom)
op_implies fof(s1_0_op_implies, axiom)
op_strict_implies fof(s1_0_op_strict_implies, axiom)
op_equiv fof(s1_0_op_equiv, axiom)
op_strict_equiv fof(s1_0_op_strict_equiv, axiom)
modus_ponens_strict_implies fof(s1_0_modus_ponens_strict_implies, axiom)
substitution_strict_equiv fof(s1_0_substitution_strict_equiv, axiom)
adjunction fof(s1_0_adjunction, axiom)
axiom_m$_1$ fof(s1_0_axiom_m$_1$, axiom)
axiom_m$_2$ fof(s1_0_axiom_m$_2$, axiom)
axiom_m$_3$ fof(s1_0_axiom_m$_3$, axiom)
axiom_m$_4$ fof(s1_0_axiom_m$_4$, axiom)
axiom_m$_5$ fof(s1_0_axiom_m$_5$, axiom)

LCL007+5.ax M6S3M9B axiomatization of S5 based on S1-0
axiom_m$_6$ fof(s1_0_m6s3m9b_axiom_m$_6$, axiom)
axiom_s$_3$ fof(s1_0_m6s3m9b_axiom_s$_3$, axiom)
axiom_m$_9$ fof(s1_0_m6s3m9b_axiom_m$_9$, axiom)
axiom_b fof(s1_0_m6s3m9b_axiom_b, axiom)

LCL007+6.ax M10 axiomatization of S5 based on S1-0
axiom_m$_{10}$ fof(s1_0_m10_axiom_m$_{10}$, axiom)

LCL009^0.ax Translating constructive S4 (CS4) to bimodal classical S4 (BS4)
reli: \$i $\to$ \$i $\to$ \$o thf(reli, type)
relr: \$i $\to$ \$i $\to$ \$o thf(relr, type)
cs4_atom: (\$i $\to$ \$o) $\to$ \$i $\to$ \$o thf(cs4_atom_decl, type)
cs4_and: (\$i $\to$ \$o) $\to$ (\$i $\to$ \$o) $\to$ \$i $\to$ \$o thf(cs4_and_decl, type)
cs4_or: (\$i $\to$ \$o) $\to$ (\$i $\to$ \$o) $\to$ \$i $\to$ \$o thf(cs4_or_decl, type)
cs4_impl: (\$i $\to$ \$o) $\to$ (\$i $\to$ \$o) $\to$ \$i $\to$ \$o thf(cs4_impl_decl, type)
cs4_true: \$i $\to$ \$o thf(cs4_true_decl, type)
cs4_false: \$i $\to$ \$o thf(cs4_false_decl, type)
cs4_all: (individuals $\to$ \$i $\to$ \$o) $\to$ \$i $\to$ \$o thf(cs4_all_decl, type)
cs4_box: (\$i $\to$ \$o) $\to$ \$i $\to$ \$o thf(cs4_box_decl, type)
cs4_atom = (λp: \$i $\to$ \$o: (mbox@reli@p)) thf(cs4_atom, definition)
cs4_and = (λa: \$i $\to$ \$o, b: \$i $\to$ \$o: (mand@a@b)) thf(cs4_and, definition)

cs4_or = (λa: $i → $o, b: $i → $o: (mor@a@b))   thf(cs4_or, definition)
cs4_impl = (λa: $i → $o, b: $i → $o: (mbox@reli@(mimpl@a@b)))   thf(cs4_impl, definition)
cs4_true = mtrue   thf(cs4_true, definition)
cs4_false = mfalse   thf(cs4_false, definition)
cs4_all = (λa: individuals → $i → $o: (mbox@reli@(mall@a)))   thf(cs4_all, definition)
cs4_box = (λa: $i → $o: (mbox@reli@(mbox@relr@a)))   thf(cs4_box, definition)
cs4_valid: ($i → $o) → $o   thf(cs4_valid_decl, type)
cs4_valid = (λa: $i → $o: (mvalid@a))   thf(cs4_valid_def, definition)
∀a: $i → $o: (mvalid@(mimpl@(mbox@reli@a)@a))   thf(refl_axiom_i, axiom)
∀a: $i → $o: (mvalid@(mimpl@(mbox@relr@a)@a))   thf(refl_axiom_r, axiom)
∀a: $i → $o: (mvalid@(mimpl@(mbox@reli@a)@(mbox@reli@(mbox@reli@a))))   thf(trans_axiom_i, axiom)
∀a: $i → $o: (mvalid@(mimpl@(mbox@relr@a)@(mbox@relr@(mbox@relr@a))))   thf(trans_axiom_r, axiom)
∀a: $i → $o: (mvalid@(mimpl@(mbox@reli@(mbox@relr@a))@(mbox@relr@(mbox@reli@a))))   thf(ax_i_r_commute, axiom

LCL013^1.ax Modal logic K
Embedding of monomodal logic K in simple type theory.
rel_k: $i → $i → $o   thf(rel_k_type, type)
mbox_k: ($i → $o) → $i → $o   thf(mbox_k_type, type)
mbox_k = (λphi: $i → $o, w: $i: ∀v: $i: (¬ rel_k@w@v or phi@v))   thf(mbox_k, definition)
mdia_k: ($i → $o) → $i → $o   thf(mdia_k_type, type)
mdia_k = (λphi: $i → $o: (mnot@(mbox_k@(mnot@phi))))   thf(mdia_k, definition)

LCL013^2.ax Modal logic D
Embedding of monomodal logic D in simple type theory
rel_d: $i → $i → $o   thf(rel_d_type, type)
mbox_d: ($i → $o) → $i → $o   thf(mbox_d_type, type)
mbox_d = (λphi: $i → $o, w: $i: ∀v: $i: (¬ rel_d@w@v or phi@v))   thf(mbox_d, definition)
mdia_d: ($i → $o) → $i → $o   thf(mdia_d_type, type)
mdia_d = (λphi: $i → $o: (mnot@(mbox_d@(mnot@phi))))   thf(mdia_d, definition)
mserial@rel_d   thf(a_1, axiom)

LCL013^3.ax Modal logic M
Embedding of monomodal logic M in simple type theory.
rel_m: $i → $i → $o   thf(rel_m_type, type)
mbox_m: ($i → $o) → $i → $o   thf(mbox_m_type, type)
mbox_m = (λphi: $i → $o, w: $i: ∀v: $i: (¬ rel_m@w@v or phi@v))   thf(mbox_m, definition)
mdia_m: ($i → $o) → $i → $o   thf(mdia_m_type, type)
mdia_m = (λphi: $i → $o: (mnot@(mbox_m@(mnot@phi))))   thf(mdia_m, definition)
mreflexive@rel_m   thf(a_1, axiom)

LCL013^4.ax Modal logic B
Embedding of monomodal logic B in simple type theory.
rel_b: $i → $i → $o   thf(rel_b_type, type)
mbox_b: ($i → $o) → $i → $o   thf(mbox_b_type, type)
mbox_b = (λphi: $i → $o, w: $i: ∀v: $i: (¬ rel_b@w@v or phi@v))   thf(mbox_b, definition)
mdia_b: ($i → $o) → $i → $o   thf(mdia_b_type, type)
mdia_b = (λphi: $i → $o: (mnot@(mbox_b@(mnot@phi))))   thf(mdia_b, definition)
mreflexive@rel_b   thf(a_1, axiom)
msymmetric@rel_b   thf(a_2, axiom)

LCL013^5.ax Modal logic S4
Embedding of monomodal logic S4 in simple type theory.
rel_s4: $i → $i → $o   thf(rel_s4_type, type)
mbox_s4: ($i → $o) → $i → $o   thf(mbox_s4_type, type)
mbox_s4 = (λphi: $i → $o, w: $i: ∀v: $i: (¬ rel_s4@w@v or phi@v))   thf(mbox_s4, definition)
mdia_s4: ($i → $o) → $i → $o   thf(mdia_s4_type, type)
mdia_s4 = (λphi: $i → $o: (mnot@(mbox_s4@(mnot@phi))))   thf(mdia_s4, definition)
mreflexive@rel_s4   thf(a_1, axiom)
mtransitive@rel_s4   thf(a_2, axiom)

LCL013^6.ax Modal logic S5
Embedding of monomodal logic S5 in simple type theory.
rel_s5: $i → $i → $o   thf(rel_s5_type, type)

mbox_s$_5$: ($i → $o) → $i → $o   thf(mbox_s5_type, type)
mbox_s$_5$ = (λphi: $i → $o, w: $i: $\forall v$: $i: ($\neg$ rel_s$_5$@w@v or phi@v))   thf(mbox_s$_5$, definition)
mdia_s$_5$: ($i → $o) → $i → $o   thf(mdia_s5_type, type)
mdia_s$_5$ = (λphi: $i → $o: (mnot@(mbox_s$_5$@(mnot@phi))))   thf(mdia_s$_5$, definition)
mreflexive@rel_s$_5$ thf(a_1, axiom)
mtransitive@rel_s$_5$ thf(a_2, axiom)
msymmetric@rel_s$_5$ thf(a_3, axiom)

LCL014^0.ax Region Connection Calculus
reg: $tType   thf(reg_type, type)
c: reg → reg → $o   thf(c_type, type)
dc: reg → reg → $o   thf(dc_type, type)
p: reg → reg → $o   thf(p_type, type)
eq: reg → reg → $o   thf(eq_type, type)
o: reg → reg → $o   thf(o_type, type)
po: reg → reg → $o   thf(po_type, type)
ec: reg → reg → $o   thf(ec_type, type)
pp: reg → reg → $o   thf(pp_type, type)
tpp: reg → reg → $o   thf(tpp_type, type)
ntpp: reg → reg → $o   thf(ntpp_type, type)
$\forall x$: reg: (c@x@x)   thf(c_reflexive, axiom)
$\forall x$: reg, y: reg: ((c@x@y) $\Rightarrow$ (c@y@x))   thf(c_symmetric, axiom)
dc = (λx: reg, y: reg: $\neg$ c@x@y)   thf(dc, definition)
p = (λx: reg, y: reg: $\forall z$: reg: ((c@z@x) $\Rightarrow$ (c@z@y)))   thf(p, definition)
eq = (λx: reg, y: reg: (p@x@y and p@y@x))   thf(eq, definition)
o = (λx: reg, y: reg: $\exists z$: reg: (p@z@x and p@z@y))   thf(o, definition)
po = (λx: reg, y: reg: (o@x@y and $\neg$ p@x@y and $\neg$ p@y@x))   thf(po, definition)
ec = (λx: reg, y: reg: (c@x@y and $\neg$ o@x@y))   thf(ec, definition)
pp = (λx: reg, y: reg: (p@x@y and $\neg$ p@y@x))   thf(pp, definition)
tpp = (λx: reg, y: reg: (pp@x@y and $\exists z$: reg: (ec@z@x and ec@z@y)))   thf(tpp, definition)
ntpp = (λx: reg, y: reg: (pp@x@y and $\neg\exists z$: reg: (ec@z@x and ec@z@y)))   thf(ntpp, definition)

LCL015^1.ax Cumulative domain specific axioms
$\forall x$: mu, v: $i, w: $i: ((exists_in_world@$x$@$v$ and rel_s$_4$@v@w) $\Rightarrow$ (exists_in_world@x@w))   thf(cumulative_ax, axiom)

LCL016^1.ax Embedding of second order modal logic in simple type theory
Extends K to KB by adding symmetric of rel.
msymmetric: ($i → $i → $o) → $o   thf(msymmetric_type, type)
msymmetric = (λr: $i → $i → $o: $\forall s$: $i, t: $i: (($r$@$s$@$t$) $\Rightarrow$ (r@t@s)))   thf(msymmetric, definition)
msymmetric@rel   thf(sym, axiom)

LCL problems

LCL001-1.p The Whitehead-Russell system => the Meredith axiom
The Whitehead-Russell axiomatisation of the Disjunction/ Negation 2 valued sentential calculus is AN-1,AN-2,AN-3, AN-4. Show that the Merideth axiom can be derived from the Whitehead-Russell axiomatisation.
(is_a_theorem(or(not(x), y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(or(not(or(not(y), z)), or(not(or(x, y)), or(x, z))))   cnf(an$_1$, axiom)
is_a_theorem(or(not(or(x, y)), or(y, x)))   cnf(an$_2$, axiom)
is_a_theorem(or(not(x), or(y, x)))   cnf(an$_3$, axiom)
is_a_theorem(or(not(or(x, x)), x))   cnf(an$_4$, axiom)
$\neg$ is_a_theorem(or(not(or(not(or(not(a), b)), or(c, or(e, falsehood)))), or(not(or(not(e), a)), or(c, or(falsehood, a)))))   cnf(pro

LCL002-1.p AN-CAMerideth => AN-1
The Whitehead-Russell axiomatisation of the Disjunction/ Negation 2 valued sentential calculus is AN-1,AN-2,AN-3, AN-4. Show that AN-1 can be derived from the Merideth axiom.
(is_a_theorem(or(not(x), y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(or(not(or(not(or(not(x), y)), or(z, or(u, v)))), or(not(or(not(u), x)), or(z, or(v, x)))))   cnf(an_CAMerideth, axi
$\neg$ is_a_theorem(or(not(or(not(b), c)), or(not(or(a, b)), or(a, c))))   cnf(an$_1$, negated_conjecture)

LCL003-1.p AN-CAMerideth => AN-2

The Whitehead-Russell axiomatisation of the Disjunction/ Negation 2 valued sentential calculus is AN-1,AN-2,AN-3, AN-4. Show that AN-2 can be derived from the Merideth axiom.
$(\text{is\_a\_theorem}(\text{or}(\text{not}(x), y)) \text{ and } \text{is\_a\_theorem}(x)) \Rightarrow \text{is\_a\_theorem}(y)$ cnf(condensed_detachment, axiom)
$\text{is\_a\_theorem}(\text{or}(\text{not}(\text{or}(\text{not}(\text{or}(\text{not}(x), y)), \text{or}(z, \text{or}(u, v)))), \text{or}(\text{not}(\text{or}(\text{not}(u), x)), \text{or}(z, \text{or}(v, x)))))$ cnf(an_CAMerideth, axi
$\neg \text{is\_a\_theorem}(\text{or}(\text{not}(\text{or}(a, b)), \text{or}(b, a)))$ cnf(an_2, negated_conjecture)

LCL004-1.p AN-CAMerideth => AN-3
The Whitehead-Russell axiomatisation of the Disjunction/ Negation 2 valued sentential calculus is AN-1,AN-2,AN-3, AN-4. Show that AN-3 can be derived from the Merideth axiom.
$(\text{is\_a\_theorem}(\text{or}(\text{not}(x), y)) \text{ and } \text{is\_a\_theorem}(x)) \Rightarrow \text{is\_a\_theorem}(y)$ cnf(condensed_detachment, axiom)
$\text{is\_a\_theorem}(\text{or}(\text{not}(\text{or}(\text{not}(\text{or}(\text{not}(x), y)), \text{or}(z, \text{or}(u, v)))), \text{or}(\text{not}(\text{or}(\text{not}(u), x)), \text{or}(z, \text{or}(v, x)))))$ cnf(an_CAMerideth, axi
$\neg \text{is\_a\_theorem}(\text{or}(\text{not}(a), \text{or}(b, a)))$ cnf(an_3, negated_conjecture)

LCL005-1.p AN-CAMerideth => AN-4
The Whitehead-Russell axiomatisation of the Disjunction/ Negation 2 valued sentential calculus is AN-1,AN-2,AN-3, AN-4. Show that AN-4 can be derived from the Merideth axiom.
$(\text{is\_a\_theorem}(\text{or}(\text{not}(x), y)) \text{ and } \text{is\_a\_theorem}(x)) \Rightarrow \text{is\_a\_theorem}(y)$ cnf(condensed_detachment, axiom)
$\text{is\_a\_theorem}(\text{or}(\text{not}(\text{or}(\text{not}(\text{or}(\text{not}(x), y)), \text{or}(z, \text{or}(u, v)))), \text{or}(\text{not}(\text{or}(\text{not}(u), x)), \text{or}(z, \text{or}(v, x)))))$ cnf(an_CAMerideth, axi
$\neg \text{is\_a\_theorem}(\text{or}(\text{not}(\text{or}(a, a)), a))$ cnf(an_4, negated_conjecture)

LCL006-1.p EC-1 depends on the Wajsberg system
Two axiomatisations of the equivalential calculus are EC-1,EC-2 by Lesniewski, and EC-4,EC-5 by Wajsburg. Show that EC-1 can be derived from the Wajsburg system.
$(\text{is\_a\_theorem}(\text{equivalent}(x, y)) \text{ and } \text{is\_a\_theorem}(x)) \Rightarrow \text{is\_a\_theorem}(y)$ cnf(condensed_detachment, axiom)
$\text{is\_a\_theorem}(\text{equivalent}(\text{equivalent}(x, y), \text{equivalent}(y, x)))$ cnf(ec_4, axiom)
$\text{is\_a\_theorem}(\text{equivalent}(\text{equivalent}(\text{equivalent}(x, y), z), \text{equivalent}(x, \text{equivalent}(y, z))))$ cnf(ec_5, axiom)
$\neg \text{is\_a\_theorem}(\text{equivalent}(\text{equivalent}(\text{equivalent}(a, b), \text{equivalent}(c, a)), \text{equivalent}(b, c)))$ cnf(prove_ec_1, negated_conjecture

LCL007-1.p EC-2 depends on the Wajsberg system
Two axiomatisations of the equivalential calculus are EC-1,EC-2 by Lesniewski, and EC-4,EC-5 by Wajsburg. Show that EC-2 can be derived from the Wajsburg system.
$(\text{is\_a\_theorem}(\text{equivalent}(x, y)) \text{ and } \text{is\_a\_theorem}(x)) \Rightarrow \text{is\_a\_theorem}(y)$ cnf(condensed_detachment, axiom)
$\text{is\_a\_theorem}(\text{equivalent}(\text{equivalent}(x, y), \text{equivalent}(y, x)))$ cnf(ec_4, axiom)
$\text{is\_a\_theorem}(\text{equivalent}(\text{equivalent}(\text{equivalent}(x, y), z), \text{equivalent}(x, \text{equivalent}(y, z))))$ cnf(ec_5, axiom)
$\neg \text{is\_a\_theorem}(\text{equivalent}(\text{equivalent}(a, \text{equivalent}(b, c)), \text{equivalent}(\text{equivalent}(a, b), c)))$ cnf(prove_ec_2, negated_conjecture

LCL008-1.p EC-4 depends on YQL
An axiomatisation of the equivalential calculus is EC-4, EC-5 by Wajsburg. Show that EC-4 can be derived from the single Lukasiewicz axiom YQL.
$(\text{is\_a\_theorem}(\text{equivalent}(x, y)) \text{ and } \text{is\_a\_theorem}(x)) \Rightarrow \text{is\_a\_theorem}(y)$ cnf(condensed_detachment, axiom)
$\text{is\_a\_theorem}(\text{equivalent}(\text{equivalent}(x, y), \text{equivalent}(\text{equivalent}(z, y), \text{equivalent}(x, z))))$ cnf(yql, axiom)
$\neg \text{is\_a\_theorem}(\text{equivalent}(\text{equivalent}(a, b), \text{equivalent}(b, a)))$ cnf(prove_ec_4, negated_conjecture)

LCL009-1.p EC-5 depends on YQL
An axiomatisation of the equivalential calculus is EC-4, EC-5 by Wajsburg. Show that EC-5 can be derived from the single Lukasiewicz axiom YQL.
$(\text{is\_a\_theorem}(\text{equivalent}(x, y)) \text{ and } \text{is\_a\_theorem}(x)) \Rightarrow \text{is\_a\_theorem}(y)$ cnf(condensed_detachment, axiom)
$\text{is\_a\_theorem}(\text{equivalent}(\text{equivalent}(x, y), \text{equivalent}(\text{equivalent}(z, y), \text{equivalent}(x, z))))$ cnf(yql, axiom)
$\neg \text{is\_a\_theorem}(\text{equivalent}(\text{equivalent}(\text{equivalent}(a, b), c), \text{equivalent}(a, \text{equivalent}(b, c))))$ cnf(prove_ec_5, negated_conjecture

LCL010-1.p YQL depends on YQF
Show that the single Lukasiewicz axiom YQL can be derived from the single Lukasiewicz axiom YQF.
$(\text{is\_a\_theorem}(\text{equivalent}(x, y)) \text{ and } \text{is\_a\_theorem}(x)) \Rightarrow \text{is\_a\_theorem}(y)$ cnf(condensed_detachment, axiom)
$\text{is\_a\_theorem}(\text{equivalent}(\text{equivalent}(x, y), \text{equivalent}(\text{equivalent}(x, z), \text{equivalent}(z, y))))$ cnf(yqf, axiom)
$\neg \text{is\_a\_theorem}(\text{equivalent}(\text{equivalent}(a, b), \text{equivalent}(\text{equivalent}(c, b), \text{equivalent}(a, c))))$ cnf(prove_yql, negated_conjecture

LCL011-1.p YQF depends on YQJ
Show that the single Lukasiewicz axiom YQF can be derived from the single Lukasiewicz axiom YQJ.
$(\text{is\_a\_theorem}(\text{equivalent}(x, y)) \text{ and } \text{is\_a\_theorem}(x)) \Rightarrow \text{is\_a\_theorem}(y)$ cnf(condensed_detachment, axiom)
$\text{is\_a\_theorem}(\text{equivalent}(\text{equivalent}(x, y), \text{equivalent}(\text{equivalent}(z, x), \text{equivalent}(y, z))))$ cnf(yqj, axiom)
$\neg \text{is\_a\_theorem}(\text{equivalent}(\text{equivalent}(a, b), \text{equivalent}(\text{equivalent}(a, c), \text{equivalent}(c, b))))$ cnf(prove_yqf, negated_conjecture

LCL012-1.p YQJ depends on UM
Show that the single Lukasiewicz axiom YQJ can be derived from the single Meredith axiom UM.
$(\text{is\_a\_theorem}(\text{equivalent}(x, y)) \text{ and } \text{is\_a\_theorem}(x)) \Rightarrow \text{is\_a\_theorem}(y)$ cnf(condensed_detachment, axiom)

is_a_theorem(equivalent(equivalent(equivalent(x, y), z), equivalent(y, equivalent(z, x)))) cnf(um, axiom)
¬ is_a_theorem(equivalent(equivalent(a, b), equivalent(equivalent(c, a), equivalent(b, c)))) cnf(prove_yqj, negated_conjecture

LCL013-1.p UM depends on XGF
Show that the single Meredith axiom UM can be derived from the single Meredith axiom XGF.
(is_a_theorem(equivalent(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(equivalent(x, equivalent(equivalent(y, equivalent(x, z)), equivalent(z, y)))) cnf(xgf, axiom)
¬ is_a_theorem(equivalent(equivalent(equivalent(a, b), c), equivalent(b, equivalent(c, a)))) cnf(prove_um, negated_conjecture

LCL014-1.p XGF depends on WN
Show that the single Meredith axiom XGF can be derived from the single Meredith axiom WN.
(is_a_theorem(equivalent(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(equivalent(equivalent(x, equivalent(y, z)), equivalent(z, equivalent(x, y)))) cnf(wn, axiom)
¬ is_a_theorem(equivalent(a, equivalent(equivalent(b, equivalent(a, c)), equivalent(c, b)))) cnf(prove_xgf, negated_conjecture

LCL015-1.p WN depends on YRM
Show that the single Meredith axiom WN can be derived from the single Meredith axiom YRM.
(is_a_theorem(equivalent(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(equivalent(equivalent(x, y), equivalent(z, equivalent(equivalent(y, z), x)))) cnf(yrm, axiom)
¬ is_a_theorem(equivalent(equivalent(a, equivalent(b, c)), equivalent(c, equivalent(a, b)))) cnf(prove_wn, negated_conjecture

LCL016-1.p YRM depends on YRO
Show that the single Meredith axiom YRM can be derived from the single Meredith axiom YRO.
(is_a_theorem(equivalent(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(equivalent(equivalent(x, y), equivalent(z, equivalent(equivalent(z, y), x)))) cnf(yro, axiom)
¬ is_a_theorem(equivalent(equivalent(a, b), equivalent(c, equivalent(equivalent(b, c), a)))) cnf(prove_yrm, negated_conjectur

LCL017-1.p YRO depends on PYO
Show that the single Meredith axiom YRO can be derived from the single Meredith axiom PYO.
(is_a_theorem(equivalent(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(equivalent(equivalent(equivalent(x, equivalent(y, z)), z), equivalent(y, x))) cnf(pyo, axiom)
¬ is_a_theorem(equivalent(equivalent(a, b), equivalent(c, equivalent(equivalent(c, b), a)))) cnf(prove_yro, negated_conjecture

LCL018-1.p PYO depends on PYM
Show that the single Meredith axiom PYO can be derived from the single Meredith axiom PYM.
(is_a_theorem(equivalent(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(equivalent(equivalent(equivalent(x, equivalent(y, z)), y), equivalent(z, x))) cnf(pym, axiom)
¬ is_a_theorem(equivalent(equivalent(equivalent(a, equivalent(b, c)), c), equivalent(b, a))) cnf(prove_pyo, negated_conjecture

LCL019-1.p PYM depends on XGK
Show that the single Meredith axiom PYM can be derived from the single Kalman axiom XGK.
(is_a_theorem(equivalent(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(equivalent(x, equivalent(equivalent(y, equivalent(z, x)), equivalent(z, y)))) cnf(xgk, axiom)
¬ is_a_theorem(equivalent(equivalent(equivalent(a, equivalent(b, c)), b), equivalent(c, a))) cnf(prove_pym, negated_conjectur

LCL020-1.p XGK depends on XHK
Show that the single Kalman axiom XGK can be derived from the single Winker axiom XHK.
(is_a_theorem(equivalent(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(equivalent(x, equivalent(equivalent(y, z), equivalent(equivalent(x, z), y)))) cnf(xhk, axiom)
¬ is_a_theorem(equivalent(a, equivalent(equivalent(b, equivalent(c, a)), equivalent(c, b)))) cnf(prove_xgk, negated_conjectur

LCL021-1.p XHK depends on XHN
Show that the single Winker axiom XHK can be derived from the single Winker axiom XHN.
(is_a_theorem(equivalent(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(equivalent(x, equivalent(equivalent(y, z), equivalent(equivalent(z, x), y)))) cnf(xhn, axiom)
¬ is_a_theorem(equivalent(a, equivalent(equivalent(b, c), equivalent(equivalent(a, c), b)))) cnf(prove_xhk, negated_conjectur

LCL022-1.p EC-1 depends on YQL
An axiomatisation of the equivalential calculus is EC-1, EC-2 by Lesniewski. Show that EC-1 can be derived from the single Lukasiewicz axiom YQL.
(is_a_theorem(equivalent(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(equivalent(equivalent(x, y), equivalent(equivalent(z, y), equivalent(x, z)))) cnf(yql, axiom)
¬ is_a_theorem(equivalent(equivalent(equivalent(a, b), equivalent(c, a)), equivalent(b, c))) cnf(prove_ec$_1$, negated_conjecture

LCL023-1.p EC-2 depends on YQL

An axiomatisation of the equivalential calculus is EC-1, EC-2 by Lesniewski. Show that EC-2 can be derived from the single Lukasiewicz axiom YQL.

(is_a_theorem(equivalent(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(equivalent(equivalent(x, y), equivalent(equivalent(z, y), equivalent(x, z))))   cnf(yql, axiom)
¬ is_a_theorem(equivalent(equivalent(a, equivalent(b, c)), equivalent(equivalent(a, b), c)))   cnf(prove_ec$_2$, negated_conjecture

LCL024-1.p PYO depends on XGK

Show that Kalman's shortest single axiom for the equivalential calculus, XGK, can be derived from the Meredith single axiom PYO.

(is_a_theorem(equivalent(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(equivalent(x, equivalent(equivalent(y, equivalent(z, x)), equivalent(z, y))))   cnf(xgk, axiom)
¬ is_a_theorem(equivalent(equivalent(equivalent(a, equivalent(b, c)), c), equivalent(b, a)))   cnf(prove_pyo, negated_conjectur

LCL025-1.p C0-1 depends on the Church system

Axiomatisations for the Implication/Falsehood 2 valued sentential calculus are C0-1,C0-2,C0-3,C0-4 by Tarski-Bernays, C0-2,C0-5,C0-6 by Church, and the single Meredith axioms. Show that C0-1 can be derived from the Church system.

(is_a_theorem(implies(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(implies(x, implies(y, x)))   cnf(c0$_2$, axiom)
is_a_theorem(implies(implies(implies(x, falsehood), falsehood), x))   cnf(c0$_5$, axiom)
is_a_theorem(implies(implies(x, implies(y, z)), implies(implies(x, y), implies(x, z))))   cnf(c0$_6$, axiom)
¬ is_a_theorem(implies(implies(a, b), implies(implies(b, c), implies(a, c))))   cnf(prove_c0$_1$, negated_conjecture)

LCL026-1.p C0-3 depends on the Church system

Axiomatisations for the Implication/Falsehood 2 valued sentential calculus are C0-1,C0-2,C0-3,C0-4 by Tarski-Bernays, C0-2,C0-5,C0-6 by Church, and the single Meredith axioms. Show that C0-3 can be derived from the Church system.

(is_a_theorem(implies(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(implies(x, implies(y, x)))   cnf(c0$_2$, axiom)
is_a_theorem(implies(implies(implies(x, falsehood), falsehood), x))   cnf(c0$_5$, axiom)
is_a_theorem(implies(implies(x, implies(y, z)), implies(implies(x, y), implies(x, z))))   cnf(c0$_6$, axiom)
¬ is_a_theorem(implies(implies(implies(a, b), a), a))   cnf(prove_c0$_3$, negated_conjecture)

LCL027-1.p C0-4 depends on the Church system

Axiomatisations for the Implication/Falsehood 2 valued sentential calculus are C0-1,C0-2,C0-3,C0-4 by Tarski-Bernays, C0-2,C0-5,C0-6 by Church, and the single Meredith axioms. Show that C0-4 can be derived from the Church system.

(is_a_theorem(implies(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(implies(x, implies(y, x)))   cnf(c0$_2$, axiom)
is_a_theorem(implies(implies(implies(x, falsehood), falsehood), x))   cnf(c0$_5$, axiom)
is_a_theorem(implies(implies(x, implies(y, z)), implies(implies(x, y), implies(x, z))))   cnf(c0$_6$, axiom)
¬ is_a_theorem(implies(falsehood, a))   cnf(prove_c0$_4$, negated_conjecture)

LCL028-1.p C0-CAMerideth depends on the Church system

Axiomatisations for the Implication/Falsehood 2 valued sentential calculus are C0-1,C0-2,C0-3,C0-4 by Tarski-Bernays, C0-2,C0-5,C0-6 by Church, and the single Meredith axioms. Show that the Meredith axiom can be derived from the Church system.

(is_a_theorem(implies(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(implies(x, implies(y, x)))   cnf(c0$_2$, axiom)
is_a_theorem(implies(implies(implies(x, falsehood), falsehood), x))   cnf(c0$_5$, axiom)
is_a_theorem(implies(implies(x, implies(y, z)), implies(implies(x, y), implies(x, z))))   cnf(c0$_6$, axiom)
¬ is_a_theorem(implies(implies(implies(implies(implies(a, b), implies(c, falsehood)), e), falsehood), implies(implies(falsehood, a)

LCL029-1.p C0-5 depends on the Tarski-Bernays system

Axiomatisations for the Implication/Falsehood 2 valued sentential calculus are C0-1,C0-2,C0-3,C0-4 by Tarski-Bernays, C0-2,C0-5,C0-6 by Church, and the single Meredith axioms. Show that C0-5 can be derived from the Tarski-Bernays system.

(is_a_theorem(implies(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z))))   cnf(c0$_1$, axiom)
is_a_theorem(implies(x, implies(y, x)))   cnf(c0$_2$, axiom)
is_a_theorem(implies(implies(implies(x, y), x), x))   cnf(c0$_3$, axiom)
is_a_theorem(implies(falsehood, x))   cnf(c0$_4$, axiom)
¬ is_a_theorem(implies(implies(implies(a, falsehood), falsehood), a))   cnf(prove_c0$_5$, negated_conjecture)

LCL030-1.p C0-6 depends on the Tarski-Bernays system
Axiomatisations for the Implication/Falsehood 2 valued sentential calculus are C0-1,C0-2,C0-3,C0-4 by Tarski-Bernays, C0-2,C0-5,C0-6 by Church, and the single Meredith axioms. Show that C0-6 can be derived from the Tarski-Bernays system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z)))) cnf(c0$_1$, axiom)
is_a_theorem(implies(x, implies(y, x))) cnf(c0$_2$, axiom)
is_a_theorem(implies(implies(implies(x, y), x), x)) cnf(c0$_3$, axiom)
is_a_theorem(implies(falsehood, x)) cnf(c0$_4$, axiom)
$\neg$ is_a_theorem(implies(implies(a, implies(b, c)), implies(implies(a, b), implies(a, c)))) cnf(prove_c0$_6$, negated_conjecture)

LCL031-1.p C0-CAMerideth depends on the Tarski-Bernays system
Axiomatisations for the Implication/Falsehood 2 valued sentential calculus are C0-1,C0-2,C0-3,C0-4 by Tarski-Bernays, C0-2,C0-5,C0-6 by Church, and the single Meredith axioms. Show that the single Meredith axiom can be derived from the Tarski-Bernays system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z)))) cnf(c0$_1$, axiom)
is_a_theorem(implies(x, implies(y, x))) cnf(c0$_2$, axiom)
is_a_theorem(implies(implies(implies(x, y), x), x)) cnf(c0$_3$, axiom)
is_a_theorem(implies(falsehood, x)) cnf(c0$_4$, axiom)
$\neg$ is_a_theorem(implies(implies(implies(implies(implies(a, b), implies(c, falsehood)), e), falsehood), implies(implies(falsehood, a)

LCL032-1.p C0-1 depends on the Merideth axiom
Axiomatisations for the Implication/Falsehood 2 valued sentential calculus are C0-1,C0-2,C0-3,C0-4 by Tarski-Bernays, C0-2,C0-5,C0-6 by Church, and the single Meredith axioms. Show that C0-1 can be derived from the single Meredith axiom.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(implies(implies(implies(x, y), implies(z, falsehood)), u), v), implies(implies(v, x), implies(z, x))))
$\neg$ is_a_theorem(implies(implies(a, b), implies(implies(b, c), implies(a, c)))) cnf(prove_c0$_1$, negated_conjecture)

LCL033-1.p C0-2 depends on the Merideth axiom
Axiomatisations for the Implication/Falsehood 2 valued sentential calculus are C0-1,C0-2,C0-3,C0-4 by Tarski-Bernays, C0-2,C0-5,C0-6 by Church, and the single Meredith axioms. Show that C0-2 can be derived from the single Meredith axiom.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(implies(implies(implies(x, y), implies(z, falsehood)), u), v), implies(implies(v, x), implies(z, x))))
$\neg$ is_a_theorem(implies(a, implies(b, a))) cnf(prove_c0$_2$, negated_conjecture)

LCL034-1.p C0-3 depends on the Merideth axiom
Axiomatisations for the Implication/Falsehood 2 valued sentential calculus are C0-1,C0-2,C0-3,C0-4 by Tarski-Bernays, C0-2,C0-5,C0-6 by Church, and the single Meredith axioms. Show that C0-3 can be derived from the single Meredith axiom.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(implies(implies(implies(x, y), implies(z, falsehood)), u), v), implies(implies(v, x), implies(z, x))))
$\neg$ is_a_theorem(implies(implies(implies(a, b), a), a)) cnf(prove_c0$_3$, negated_conjecture)

LCL035-1.p C0-4 depends on the Merideth axiom
Axiomatisations for the Implication/Falsehood 2 valued sentential calculus are C0-1,C0-2,C0-3,C0-4 by Tarski-Bernays, C0-2,C0-5,C0-6 by Church, and the single Meredith axioms. Show that C0-4 can be derived from the single Meredith axiom.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(implies(implies(implies(x, y), implies(z, falsehood)), u), v), implies(implies(v, x), implies(z, x))))
$\neg$ is_a_theorem(implies(falsehood, a)) cnf(prove_c0$_4$, negated_conjecture)

LCL036-1.p C0-5 depends on the Merideth axiom
Axiomatisations for the Implication/Falsehood 2 valued sentential calculus are C0-1,C0-2,C0-3,C0-4 by Tarski-Bernays, C0-2,C0-5,C0-6 by Church, and the single Meredith axioms. Show that C0-5 can be derived from the single Meredith axiom.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(implies(implies(implies(x, y), implies(z, falsehood)), u), v), implies(implies(v, x), implies(z, x))))
$\neg$ is_a_theorem(implies(implies(implies(a, falsehood), falsehood), a)) cnf(prove_c0$_5$, negated_conjecture)

LCL037-1.p C0-6 depends on the Merideth axiom

Axiomatisations for the Implication/Falsehood 2 valued sentential calculus are C0-1,C0-2,C0-3,C0-4 by Tarski-Bernays, C0-2,C0-5,C0-6 by Church, and the single Meredith axioms. Show that C0-6 can be derived from the single Meredith axiom.

(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(implies(implies(implies(x, y), implies(z, falsehood)), u), v), implies(implies(v, x), implies(z, x)))) cnf(single_axiom, axiom)
$\neg$ is_a_theorem(implies(implies(a, implies(b, c)), implies(implies(a, b), implies(a, c)))) cnf(prove_c0$_6$, negated_conjecture)

LCL038-1.p C0-1 depends on a single axiom
An axiomatisation for the Implication/Falsehood 2 valued sentential calculus is C0-1,C0-2,C0-3,C0-4 by Tarski-Bernays. Show that C0-1 can be derived from the first Lukasiewicz axiom.

(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(implies(x, y), z), implies(implies(z, x), implies(u, x)))) cnf(ic_JLukasiewicz, axiom)
$\neg$ is_a_theorem(implies(implies(a, b), implies(implies(b, c), implies(a, c)))) cnf(prove_c0$_1$, negated_conjecture)

LCL039-1.p A theorem from Morgan
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-18,CN-35,CN-49 by Church. This can be extended to the modal logic T by the addition of three axioms for the modal operators. This problem proves a simple result of T.

(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(x, implies(y, x))) cnf(cn$_{18}$, axiom)
is_a_theorem(implies(implies(x, implies(y, z)), implies(implies(x, y), implies(x, z)))) cnf(cn$_{35}$, axiom)
is_a_theorem(implies(implies(not(x), not(y)), implies(y, x))) cnf(cn$_{49}$, axiom)
is_a_theorem(implies(necessary(implies(x, y)), implies(necessary(x), necessary(y)))) cnf(necessitation$_1$, axiom)
is_a_theorem(implies(necessary(x), x)) cnf(necessitation$_2$, axiom)
is_a_theorem(x) $\Rightarrow$ is_a_theorem(necessary(x)) cnf(axiom_of_necessitation, axiom)
$\neg$ is_a_theorem(implies(necessary(a), not(necessary(not(a))))) cnf(prove_this, negated_conjecture)

LCL040-1.p CN-21 depends on the rest of Frege's system
The first axiomatisation of Implication/Negation 2 valued sentential calculus was CN-18,CN-21,CN-35,CN-39,CN-39, CN-40,CN-46 by Frege. Show, like Lukasiewicz did, that CN-21 depends on the rest of this axiomatisation.

(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(x, implies(y, x))) cnf(cn$_{18}$, axiom)
is_a_theorem(implies(implies(x, implies(y, z)), implies(implies(x, y), implies(x, z)))) cnf(cn$_{35}$, axiom)
is_a_theorem(implies(not(not(x)), x)) cnf(cn$_{39}$, axiom)
is_a_theorem(implies(x, not(not(x)))) cnf(cn$_{40}$, axiom)
is_a_theorem(implies(implies(x, y), implies(not(y), not(x)))) cnf(cn$_{46}$, axiom)
$\neg$ is_a_theorem(implies(implies(a, implies(b, c)), implies(b, implies(a, c)))) cnf(prove_cn$_{21}$, negated_conjecture)

LCL041-1.p CN-30 depends on the rest of Hilbert's system
An early axiomatisation of Implication/Negation 2 valued sentential calculus was CN-3,CN-18,CN-21,CN-22,CN-30, CN-54 by Hilbert. Show, like Lukasiewicz did, that CN-30 depends on the rest of this axiomatisation.

(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(x, implies(not(x), y))) cnf(cn$_3$, axiom)
is_a_theorem(implies(x, implies(y, x))) cnf(cn$_{18}$, axiom)
is_a_theorem(implies(implies(x, implies(y, z)), implies(y, implies(x, z)))) cnf(cn$_{21}$, axiom)
is_a_theorem(implies(implies(y, z), implies(implies(x, y), implies(x, z)))) cnf(cn$_{22}$, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(not(x), y), y))) cnf(cn$_{54}$, axiom)
$\neg$ is_a_theorem(implies(implies(a, implies(a, b)), implies(a, b))) cnf(prove_cn$_{30}$, negated_conjecture)

LCL042-1.p CN-35 depends on Hilbert's system
Two axiomatisations of the Implication/Negation 2 valued sentential calculus are CN-18,CN-21,CN-35,CN-39,CN-39, CN-40,CN-46 by Frege and CN-3,CN-18,CN-21,CN-22,CN-30, CN-54 by Hilbert. Show that CN-35 depends on the simplified Hilbert system (without CN-30).

(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(x, implies(not(x), y))) cnf(cn$_3$, axiom)
is_a_theorem(implies(x, implies(y, x))) cnf(cn$_{18}$, axiom)
is_a_theorem(implies(implies(x, implies(y, z)), implies(y, implies(x, z)))) cnf(cn$_{21}$, axiom)
is_a_theorem(implies(implies(y, z), implies(implies(x, y), implies(x, z)))) cnf(cn$_{22}$, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(not(x), y), y))) cnf(cn$_{54}$, axiom)
$\neg$ is_a_theorem(implies(implies(a, implies(b, c)), implies(implies(a, b), implies(a, c)))) cnf(prove_cn$_{35}$, negated_conjecture)

LCL043-1.p CN-39 depends on Hilbert's system

Two axiomatisations of the Implication/Negation 2 valued sentential calculus are CN-18,CN-21,CN-35,CN-39,CN-39, CN-40,CN-46 by Frege and CN-3,CN-18,CN-21,CN-22,CN-30, CN-54 by Hilbert. Show that CN-39 depends on the simplified Hilbert system (without CN-30).

$(\text{is\_a\_theorem}(\text{implies}(x, y)) \text{ and } \text{is\_a\_theorem}(x)) \Rightarrow \text{is\_a\_theorem}(y) \quad \text{cnf}(\text{condensed\_detachment}, \text{axiom})$

$\text{is\_a\_theorem}(\text{implies}(x, \text{implies}(\text{not}(x), y))) \quad \text{cnf}(\text{cn}_3, \text{axiom})$

$\text{is\_a\_theorem}(\text{implies}(x, \text{implies}(y, x))) \quad \text{cnf}(\text{cn}_{18}, \text{axiom})$

$\text{is\_a\_theorem}(\text{implies}(\text{implies}(x, \text{implies}(y, z)), \text{implies}(y, \text{implies}(x, z)))) \quad \text{cnf}(\text{cn}_{21}, \text{axiom})$

$\text{is\_a\_theorem}(\text{implies}(\text{implies}(y, z), \text{implies}(\text{implies}(x, y), \text{implies}(x, z)))) \quad \text{cnf}(\text{cn}_{22}, \text{axiom})$

$\text{is\_a\_theorem}(\text{implies}(\text{implies}(x, y), \text{implies}(\text{implies}(\text{not}(x), y), y))) \quad \text{cnf}(\text{cn}_{54}, \text{axiom})$

$\neg \text{is\_a\_theorem}(\text{implies}(\text{not}(\text{not}(a)), a)) \quad \text{cnf}(\text{prove\_cn}_{39}, \text{negated\_conjecture})$

LCL044-1.p CN-40 depends on Hilbert's system

Two axiomatisations of the Implication/Negation 2 valued sentential calculus are CN-18,CN-21,CN-35,CN-39,CN-39, CN-40,CN-46 by Frege and CN-3,CN-18,CN-21,CN-22,CN-30, CN-54 by Hilbert. Show that CN-40 depends on the simplified Hilbert system (without CN-30).

$(\text{is\_a\_theorem}(\text{implies}(x, y)) \text{ and } \text{is\_a\_theorem}(x)) \Rightarrow \text{is\_a\_theorem}(y) \quad \text{cnf}(\text{condensed\_detachment}, \text{axiom})$

$\text{is\_a\_theorem}(\text{implies}(x, \text{implies}(\text{not}(x), y))) \quad \text{cnf}(\text{cn}_3, \text{axiom})$

$\text{is\_a\_theorem}(\text{implies}(x, \text{implies}(y, x))) \quad \text{cnf}(\text{cn}_{18}, \text{axiom})$

$\text{is\_a\_theorem}(\text{implies}(\text{implies}(x, \text{implies}(y, z)), \text{implies}(y, \text{implies}(x, z)))) \quad \text{cnf}(\text{cn}_{21}, \text{axiom})$

$\text{is\_a\_theorem}(\text{implies}(\text{implies}(y, z), \text{implies}(\text{implies}(x, y), \text{implies}(x, z)))) \quad \text{cnf}(\text{cn}_{22}, \text{axiom})$

$\text{is\_a\_theorem}(\text{implies}(\text{implies}(x, y), \text{implies}(\text{implies}(\text{not}(x), y), y))) \quad \text{cnf}(\text{cn}_{54}, \text{axiom})$

$\neg \text{is\_a\_theorem}(\text{implies}(a, \text{not}(\text{not}(a)))) \quad \text{cnf}(\text{prove\_cn}_{40}, \text{negated\_conjecture})$

LCL045-1.p CN-46 depends on Hilbert's system

Two axiomatisations of the Implication/Negation 2 valued sentential calculus are CN-18,CN-21,CN-35,CN-39,CN-39, CN-40,CN-46 by Frege and CN-3,CN-18,CN-21,CN-22,CN-30, CN-54 by Hilbert. Show that CN-46 depends on the simplified Hilbert system (without CN-30).

$(\text{is\_a\_theorem}(\text{implies}(x, y)) \text{ and } \text{is\_a\_theorem}(x)) \Rightarrow \text{is\_a\_theorem}(y) \quad \text{cnf}(\text{condensed\_detachment}, \text{axiom})$

$\text{is\_a\_theorem}(\text{implies}(x, \text{implies}(\text{not}(x), y))) \quad \text{cnf}(\text{cn}_3, \text{axiom})$

$\text{is\_a\_theorem}(\text{implies}(x, \text{implies}(y, x))) \quad \text{cnf}(\text{cn}_{18}, \text{axiom})$

$\text{is\_a\_theorem}(\text{implies}(\text{implies}(x, \text{implies}(y, z)), \text{implies}(y, \text{implies}(x, z)))) \quad \text{cnf}(\text{cn}_{21}, \text{axiom})$

$\text{is\_a\_theorem}(\text{implies}(\text{implies}(y, z), \text{implies}(\text{implies}(x, y), \text{implies}(x, z)))) \quad \text{cnf}(\text{cn}_{22}, \text{axiom})$

$\text{is\_a\_theorem}(\text{implies}(\text{implies}(x, y), \text{implies}(\text{implies}(\text{not}(x), y), y))) \quad \text{cnf}(\text{cn}_{54}, \text{axiom})$

$\neg \text{is\_a\_theorem}(\text{implies}(\text{implies}(a, b), \text{implies}(\text{not}(b), \text{not}(a)))) \quad \text{cnf}(\text{prove\_cn}_{46}, \text{negated\_conjecture})$

LCL046-1.p CN-16 depends on the Lukasiewicz system

Axiomatisations of the Implication/Negation 2 valued sentential calculus are CN-1,CN-2,CN-3 by Lukasiewicz, CN-18,CN-21,CN-35,CN-39,CN-39,CN-40,CN-46 by Frege, CN-3,CN-18,CN-21,CN-22,CN-30,CN-54 by Hilbert, CN-18, CN-35,CN-49 by Church, CN-19,CN-37,CN-59 by Lukasiewicz, CN-19,CN-37,CN-60 by Wos, and the single Meredith axiom. Show that CN-16 depends on the short Lukasiewicz system.

$(\text{is\_a\_theorem}(\text{implies}(x, y)) \text{ and } \text{is\_a\_theorem}(x)) \Rightarrow \text{is\_a\_theorem}(y) \quad \text{cnf}(\text{condensed\_detachment}, \text{axiom})$

$\text{is\_a\_theorem}(\text{implies}(\text{implies}(x, y), \text{implies}(\text{implies}(y, z), \text{implies}(x, z)))) \quad \text{cnf}(\text{cn}_1, \text{axiom})$

$\text{is\_a\_theorem}(\text{implies}(\text{implies}(\text{not}(x), x), x)) \quad \text{cnf}(\text{cn}_2, \text{axiom})$

$\text{is\_a\_theorem}(\text{implies}(x, \text{implies}(\text{not}(x), y))) \quad \text{cnf}(\text{cn}_3, \text{axiom})$

$\neg \text{is\_a\_theorem}(\text{implies}(a, a)) \quad \text{cnf}(\text{prove\_cn}_{16}, \text{negated\_conjecture})$

LCL047-1.p CN-18 depends on the Lukasiewicz system

Axiomatisations of the Implication/Negation 2 valued sentential calculus are CN-1,CN-2,CN-3 by Lukasiewicz, CN-18,CN-21,CN-35,CN-39,CN-39,CN-40,CN-46 by Frege, CN-3,CN-18,CN-21,CN-22,CN-30,CN-54 by Hilbert, CN-18, CN-35,CN-49 by Church, CN-19,CN-37,CN-59 by Lukasiewicz, CN-19,CN-37,CN-60 by Wos, and the single Meredith axiom. Show that CN-18 depends on the short Lukasiewicz system.

$(\text{is\_a\_theorem}(\text{implies}(x, y)) \text{ and } \text{is\_a\_theorem}(x)) \Rightarrow \text{is\_a\_theorem}(y) \quad \text{cnf}(\text{condensed\_detachment}, \text{axiom})$

$\text{is\_a\_theorem}(\text{implies}(\text{implies}(x, y), \text{implies}(\text{implies}(y, z), \text{implies}(x, z)))) \quad \text{cnf}(\text{cn}_1, \text{axiom})$

$\text{is\_a\_theorem}(\text{implies}(\text{implies}(\text{not}(x), x), x)) \quad \text{cnf}(\text{cn}_2, \text{axiom})$

$\text{is\_a\_theorem}(\text{implies}(x, \text{implies}(\text{not}(x), y))) \quad \text{cnf}(\text{cn}_3, \text{axiom})$

$\neg \text{is\_a\_theorem}(\text{implies}(a, \text{implies}(b, a))) \quad \text{cnf}(\text{prove\_cn}_{18}, \text{negated\_conjecture})$

LCL048-1.p CN-19 depends on the Lukasiewicz system

Axiomatisations of the Implication/Negation 2 valued sentential calculus are CN-1,CN-2,CN-3 by Lukasiewicz, CN-18,CN-21,CN-35,CN-39,CN-39,CN-40,CN-46 by Frege, CN-3,CN-18,CN-21,CN-22,CN-30,CN-54 by Hilbert, CN-18, CN-35,CN-49 by Church, CN-19,CN-37,CN-59 by Lukasiewicz, CN-19,CN-37,CN-60 by Wos, and the single Meredith axiom. Show that CN-19 depends on the short Lukasiewicz system.

(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z)))) cnf(cn_1, axiom)
is_a_theorem(implies(implies(not(x), x), x)) cnf(cn_2, axiom)
is_a_theorem(implies(x, implies(not(x), y))) cnf(cn_3, axiom)
$\neg$ is_a_theorem(implies(implies(implies(a, b), c), implies(b, c))) cnf(prove_cn_{19}, negated_conjecture)

LCL049-1.p CN-20 depends on the Lukasiewicz system
Axiomatisations of the Implication/Negation 2 valued sentential calculus are CN-1,CN-2,CN-3 by Lukasiewicz, CN-18,CN-21,CN-35,CN-39,CN-39,CN-40,CN-46 by Frege, CN-3,CN-18,CN-21,CN-22,CN-30,CN-54 by Hilbert, CN-18, CN-35,CN-49 by Church, CN-19,CN-37,CN-59 by Lukasiewicz, CN-19,CN-37,CN-60 by Wos, and the single Meredith axiom. Show that CN-20 depends on the short Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z)))) cnf(cn_1, axiom)
is_a_theorem(implies(implies(not(x), x), x)) cnf(cn_2, axiom)
is_a_theorem(implies(x, implies(not(x), y))) cnf(cn_3, axiom)
$\neg$ is_a_theorem(implies(a, implies(implies(a, b), b))) cnf(prove_cn_{20}, negated_conjecture)

LCL050-1.p CN-21 depends on the Lukasiewicz system
Axiomatisations of the Implication/Negation 2 valued sentential calculus are CN-1,CN-2,CN-3 by Lukasiewicz, CN-18,CN-21,CN-35,CN-39,CN-39,CN-40,CN-46 by Frege, CN-3,CN-18,CN-21,CN-22,CN-30,CN-54 by Hilbert, CN-18, CN-35,CN-49 by Church, CN-19,CN-37,CN-59 by Lukasiewicz, CN-19,CN-37,CN-60 by Wos, and the single Meredith axiom. Show that CN-21 depends on the short Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z)))) cnf(cn_1, axiom)
is_a_theorem(implies(implies(not(x), x), x)) cnf(cn_2, axiom)
is_a_theorem(implies(x, implies(not(x), y))) cnf(cn_3, axiom)
$\neg$ is_a_theorem(implies(implies(a, implies(b, c)), implies(b, implies(a, c)))) cnf(prove_cn_{21}, negated_conjecture)

LCL051-1.p CN-22 depends on the Lukasiewicz system
Axiomatisations of the Implication/Negation 2 valued sentential calculus are CN-1,CN-2,CN-3 by Lukasiewicz, CN-18,CN-21,CN-35,CN-39,CN-39,CN-40,CN-46 by Frege, CN-3,CN-18,CN-21,CN-22,CN-30,CN-54 by Hilbert, CN-18, CN-35,CN-49 by Church, CN-19,CN-37,CN-59 by Lukasiewicz, CN-19,CN-37,CN-60 by Wos, and the single Meredith axiom. Show that CN-22 depends on the short Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z)))) cnf(cn_1, axiom)
is_a_theorem(implies(implies(not(x), x), x)) cnf(cn_2, axiom)
is_a_theorem(implies(x, implies(not(x), y))) cnf(cn_3, axiom)
$\neg$ is_a_theorem(implies(implies(b, c), implies(implies(a, b), implies(a, c)))) cnf(prove_cn_{22}, negated_conjecture)

LCL052-1.p CN-24 depends on the Lukasiewicz system
Axiomatisations of the Implication/Negation 2 valued sentential calculus are CN-1,CN-2,CN-3 by Lukasiewicz, CN-18,CN-21,CN-35,CN-39,CN-39,CN-40,CN-46 by Frege, CN-3,CN-18,CN-21,CN-22,CN-30,CN-54 by Hilbert, CN-18, CN-35,CN-49 by Church, CN-19,CN-37,CN-59 by Lukasiewicz, CN-19,CN-37,CN-60 by Wos, and the single Meredith axiom. Show that CN-24 depends on the short Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z)))) cnf(cn_1, axiom)
is_a_theorem(implies(implies(not(x), x), x)) cnf(cn_2, axiom)
is_a_theorem(implies(x, implies(not(x), y))) cnf(cn_3, axiom)
$\neg$ is_a_theorem(implies(implies(implies(a, b), a), a)) cnf(prove_cn_{24}, negated_conjecture)

LCL053-1.p CN-30 depends on the Lukasiewicz system
Axiomatisations of the Implication/Negation 2 valued sentential calculus are CN-1,CN-2,CN-3 by Lukasiewicz, CN-18,CN-21,CN-35,CN-39,CN-39,CN-40,CN-46 by Frege, CN-3,CN-18,CN-21,CN-22,CN-30,CN-54 by Hilbert, CN-18, CN-35,CN-49 by Church, CN-19,CN-37,CN-59 by Lukasiewicz, CN-19,CN-37,CN-60 by Wos, and the single Meredith axiom. Show that CN-30 depends on the short Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z)))) cnf(cn_1, axiom)
is_a_theorem(implies(implies(not(x), x), x)) cnf(cn_2, axiom)
is_a_theorem(implies(x, implies(not(x), y))) cnf(cn_3, axiom)
$\neg$ is_a_theorem(implies(implies(a, implies(a, b)), implies(a, b))) cnf(prove_cn_{30}, negated_conjecture)

LCL054-1.p CN-35 depends on the Lukasiewicz system

Axiomatisations of the Implication/Negation 2 valued sentential calculus are CN-1,CN-2,CN-3 by Lukasiewicz, CN-18,CN-21,CN-35,CN-39,CN-39,CN-40,CN-46 by Frege, CN-3,CN-18,CN-21,CN-22,CN-30,CN-54 by Hilbert, CN-18, CN-35,CN-49 by Church, CN-19,CN-37,CN-59 by Lukasiewicz, CN-19,CN-37,CN-60 by Wos, and the single Meredith axiom. Show that CN-35 depends on the short Lukasiewicz system.
$(\text{is\_a\_theorem}(\text{implies}(x, y)) \text{ and } \text{is\_a\_theorem}(x)) \Rightarrow \text{is\_a\_theorem}(y)$ cnf(condensed_detachment, axiom)
$\text{is\_a\_theorem}(\text{implies}(\text{implies}(x, y), \text{implies}(\text{implies}(y, z), \text{implies}(x, z))))$ cnf(cn_1, axiom)
$\text{is\_a\_theorem}(\text{implies}(\text{implies}(\text{not}(x), x), x))$ cnf(cn_2, axiom)
$\text{is\_a\_theorem}(\text{implies}(x, \text{implies}(\text{not}(x), y)))$ cnf(cn_3, axiom)
$\neg\,\text{is\_a\_theorem}(\text{implies}(\text{implies}(a, \text{implies}(b, c)), \text{implies}(\text{implies}(a, b), \text{implies}(a, c))))$ cnf(prove_cn_{35}, negated_conjecture)

LCL055-1.p CN-37 depends on the Lukasiewicz system
Axiomatisations of the Implication/Negation 2 valued sentential calculus are CN-1,CN-2,CN-3 by Lukasiewicz, CN-18,CN-21,CN-35,CN-39,CN-39,CN-40,CN-46 by Frege, CN-3,CN-18,CN-21,CN-22,CN-30,CN-54 by Hilbert, CN-18, CN-35,CN-49 by Church, CN-19,CN-37,CN-59 by Lukasiewicz, CN-19,CN-37,CN-60 by Wos, and the single Meredith axiom. Show that CN-37 depends on the short Lukasiewicz system.
$(\text{is\_a\_theorem}(\text{implies}(x, y)) \text{ and } \text{is\_a\_theorem}(x)) \Rightarrow \text{is\_a\_theorem}(y)$ cnf(condensed_detachment, axiom)
$\text{is\_a\_theorem}(\text{implies}(\text{implies}(x, y), \text{implies}(\text{implies}(y, z), \text{implies}(x, z))))$ cnf(cn_1, axiom)
$\text{is\_a\_theorem}(\text{implies}(\text{implies}(\text{not}(x), x), x))$ cnf(cn_2, axiom)
$\text{is\_a\_theorem}(\text{implies}(x, \text{implies}(\text{not}(x), y)))$ cnf(cn_3, axiom)
$\neg\,\text{is\_a\_theorem}(\text{implies}(\text{implies}(\text{implies}(a, b), c), \text{implies}(\text{not}(a), c)))$ cnf(prove_cn_{37}, negated_conjecture)

LCL056-1.p CN-39 depends on the Lukasiewicz system
Axiomatisations of the Implication/Negation 2 valued sentential calculus are CN-1,CN-2,CN-3 by Lukasiewicz, CN-18,CN-21,CN-35,CN-39,CN-39,CN-40,CN-46 by Frege, CN-3,CN-18,CN-21,CN-22,CN-30,CN-54 by Hilbert, CN-18, CN-35,CN-49 by Church, CN-19,CN-37,CN-59 by Lukasiewicz, CN-19,CN-37,CN-60 by Wos, and the single Meredith axiom. Show that CN-39 depends on the short Lukasiewicz system.
$(\text{is\_a\_theorem}(\text{implies}(x, y)) \text{ and } \text{is\_a\_theorem}(x)) \Rightarrow \text{is\_a\_theorem}(y)$ cnf(condensed_detachment, axiom)
$\text{is\_a\_theorem}(\text{implies}(\text{implies}(x, y), \text{implies}(\text{implies}(y, z), \text{implies}(x, z))))$ cnf(cn_1, axiom)
$\text{is\_a\_theorem}(\text{implies}(\text{implies}(\text{not}(x), x), x))$ cnf(cn_2, axiom)
$\text{is\_a\_theorem}(\text{implies}(x, \text{implies}(\text{not}(x), y)))$ cnf(cn_3, axiom)
$\neg\,\text{is\_a\_theorem}(\text{implies}(\text{not}(\text{not}(a)), a))$ cnf(prove_cn_{39}, negated_conjecture)

LCL057-1.p CN-40 depends on the Lukasiewicz system
Axiomatisations of the Implication/Negation 2 valued sentential calculus are CN-1,CN-2,CN-3 by Lukasiewicz, CN-18,CN-21,CN-35,CN-39,CN-39,CN-40,CN-46 by Frege, CN-3,CN-18,CN-21,CN-22,CN-30,CN-54 by Hilbert, CN-18, CN-35,CN-49 by Church, CN-19,CN-37,CN-59 by Lukasiewicz, CN-19,CN-37,CN-60 by Wos, and the single Meredith axiom. Show that CN-40 depends on the short Lukasiewicz system.
$(\text{is\_a\_theorem}(\text{implies}(x, y)) \text{ and } \text{is\_a\_theorem}(x)) \Rightarrow \text{is\_a\_theorem}(y)$ cnf(condensed_detachment, axiom)
$\text{is\_a\_theorem}(\text{implies}(\text{implies}(x, y), \text{implies}(\text{implies}(y, z), \text{implies}(x, z))))$ cnf(cn_1, axiom)
$\text{is\_a\_theorem}(\text{implies}(\text{implies}(\text{not}(x), x), x))$ cnf(cn_2, axiom)
$\text{is\_a\_theorem}(\text{implies}(x, \text{implies}(\text{not}(x), y)))$ cnf(cn_3, axiom)
$\neg\,\text{is\_a\_theorem}(\text{implies}(a, \text{not}(\text{not}(a))))$ cnf(prove_cn_{40}, negated_conjecture)

LCL058-1.p CN-46 depends on the Lukasiewicz system
Axiomatisations of the Implication/Negation 2 valued sentential calculus are CN-1,CN-2,CN-3 by Lukasiewicz, CN-18,CN-21,CN-35,CN-39,CN-39,CN-40,CN-46 by Frege, CN-3,CN-18,CN-21,CN-22,CN-30,CN-54 by Hilbert, CN-18, CN-35,CN-49 by Church, CN-19,CN-37,CN-59 by Lukasiewicz, CN-19,CN-37,CN-60 by Wos, and the single Meredith axiom. Show that CN-46 depends on the short Lukasiewicz system.
$(\text{is\_a\_theorem}(\text{implies}(x, y)) \text{ and } \text{is\_a\_theorem}(x)) \Rightarrow \text{is\_a\_theorem}(y)$ cnf(condensed_detachment, axiom)
$\text{is\_a\_theorem}(\text{implies}(\text{implies}(x, y), \text{implies}(\text{implies}(y, z), \text{implies}(x, z))))$ cnf(cn_1, axiom)
$\text{is\_a\_theorem}(\text{implies}(\text{implies}(\text{not}(x), x), x))$ cnf(cn_2, axiom)
$\text{is\_a\_theorem}(\text{implies}(x, \text{implies}(\text{not}(x), y)))$ cnf(cn_3, axiom)
$\neg\,\text{is\_a\_theorem}(\text{implies}(\text{implies}(a, b), \text{implies}(\text{not}(b), \text{not}(a))))$ cnf(prove_cn_{46}, negated_conjecture)

LCL059-1.p CN-49 depends on the Lukasiewicz system
Axiomatisations of the Implication/Negation 2 valued sentential calculus are CN-1,CN-2,CN-3 by Lukasiewicz, CN-18,CN-21,CN-35,CN-39,CN-39,CN-40,CN-46 by Frege, CN-3,CN-18,CN-21,CN-22,CN-30,CN-54 by Hilbert, CN-18, CN-35,CN-49 by Church, CN-19,CN-37,CN-59 by Lukasiewicz, CN-19,CN-37,CN-60 by Wos, and the single Meredith axiom. Show that CN-49 depends on the short Lukasiewicz system.
$(\text{is\_a\_theorem}(\text{implies}(x, y)) \text{ and } \text{is\_a\_theorem}(x)) \Rightarrow \text{is\_a\_theorem}(y)$ cnf(condensed_detachment, axiom)
$\text{is\_a\_theorem}(\text{implies}(\text{implies}(x, y), \text{implies}(\text{implies}(y, z), \text{implies}(x, z))))$ cnf(cn_1, axiom)
$\text{is\_a\_theorem}(\text{implies}(\text{implies}(\text{not}(x), x), x))$ cnf(cn_2, axiom)

is_a_theorem(implies(x, implies(not(x), y)))   cnf(cn$_{3}$, axiom)
¬ is_a_theorem(implies(implies(not(a), not(b)), implies(b, a)))   cnf(prove_cn$_{49}$, negated_conjecture)

LCL060-1.p CN-54 depends on the Lukasiewicz system
Axiomatisations of the Implication/Negation 2 valued sentential calculus are CN-1,CN-2,CN-3 by Lukasiewicz, CN-18,CN-21,CN-35,CN-39,CN-39,CN-40,CN-46 by Frege, CN-3,CN-18,CN-21,CN-22,CN-30,CN-54 by Hilbert, CN-18, CN-35,CN-49 by Church, CN-19,CN-37,CN-59 by Lukasiewicz, CN-19,CN-37,CN-60 by Wos, and the single Meredith axiom. Show that CN-54 depends on the short Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z))))   cnf(cn$_{1}$, axiom)
is_a_theorem(implies(implies(not(x), x), x))   cnf(cn$_{2}$, axiom)
is_a_theorem(implies(x, implies(not(x), y)))   cnf(cn$_{3}$, axiom)
¬ is_a_theorem(implies(implies(a, b), implies(implies(not(a), b), b)))   cnf(prove_cn$_{54}$, negated_conjecture)

LCL061-1.p CN-59 depends on the Lukasiewicz system
Axiomatisations of the Implication/Negation 2 valued sentential calculus are CN-1,CN-2,CN-3 by Lukasiewicz, CN-18,CN-21,CN-35,CN-39,CN-39,CN-40,CN-46 by Frege, CN-3,CN-18,CN-21,CN-22,CN-30,CN-54 by Hilbert, CN-18, CN-35,CN-49 by Church, CN-19,CN-37,CN-59 by Lukasiewicz, CN-19,CN-37,CN-60 by Wos, and the single Meredith axiom. Show that CN-59 depends on the short Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z))))   cnf(cn$_{1}$, axiom)
is_a_theorem(implies(implies(not(x), x), x))   cnf(cn$_{2}$, axiom)
is_a_theorem(implies(x, implies(not(x), y)))   cnf(cn$_{3}$, axiom)
¬ is_a_theorem(implies(implies(not(a), c), implies(implies(b, c), implies(implies(a, b), c))))   cnf(prove_cn$_{59}$, negated_conjectu

LCL062-1.p CN-60 depends on the Lukasiewicz system
Axiomatisations of the Implication/Negation 2 valued sentential calculus are CN-1,CN-2,CN-3 by Lukasiewicz, CN-18,CN-21,CN-35,CN-39,CN-39,CN-40,CN-46 by Frege, CN-3,CN-18,CN-21,CN-22,CN-30,CN-54 by Hilbert, CN-18, CN-35,CN-49 by Church, CN-19,CN-37,CN-59 by Lukasiewicz, CN-19,CN-37,CN-60 by Wos, and the single Meredith axiom. Show that CN-60 depends on the short Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z))))   cnf(cn$_{1}$, axiom)
is_a_theorem(implies(implies(not(x), x), x))   cnf(cn$_{2}$, axiom)
is_a_theorem(implies(x, implies(not(x), y)))   cnf(cn$_{3}$, axiom)
¬ is_a_theorem(implies(implies(a, implies(not(b), c)), implies(a, implies(implies(e, c), implies(implies(b, e), c)))))   cnf(prove_c

LCL063-1.p CN-CAMerideth depends on the Lukasiewicz system
Axiomatisations of the Implication/Negation 2 valued sentential calculus are CN-1,CN-2,CN-3 by Lukasiewicz, CN-18,CN-21,CN-35,CN-39,CN-39,CN-40,CN-46 by Frege, CN-3,CN-18,CN-21,CN-22,CN-30,CN-54 by Hilbert, CN-18, CN-35,CN-49 by Church, CN-19,CN-37,CN-59 by Lukasiewicz, CN-19,CN-37,CN-60 by Wos, and the single Meredith axiom. Show that the single Meredith axiom depends on the short Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z))))   cnf(cn$_{1}$, axiom)
is_a_theorem(implies(implies(not(x), x), x))   cnf(cn$_{2}$, axiom)
is_a_theorem(implies(x, implies(not(x), y)))   cnf(cn$_{3}$, axiom)
¬ is_a_theorem(implies(implies(implies(implies(implies(a, b), implies(not(c), not(e))), c), falsehood), implies(implies(falsehood,

LCL064-1.p CN-1 depends on the Church system
Axiomatisations of the Implication/Negation 2 valued sentential calculus are CN-1,CN-2,CN-3 by Lukasiewicz, CN-18,CN-21,CN-35,CN-39,CN-39,CN-40,CN-46 by Frege, CN-3,CN-18,CN-21,CN-22,CN-30,CN-54 by Hilbert, CN-18, CN-35,CN-49 by Church, CN-19,CN-37,CN-59 by Lukasiewicz, CN-19,CN-37,CN-60 by Wos, and the single Meredith axiom. Show that CN-X depends on the Church system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(implies(x, implies(y, x)))   cnf(cn$_{18}$, axiom)
is_a_theorem(implies(implies(x, implies(y, z)), implies(implies(x, y), implies(x, z))))   cnf(cn$_{35}$, axiom)
is_a_theorem(implies(implies(not(x), not(y)), implies(y, x)))   cnf(cn$_{49}$, axiom)
¬ is_a_theorem(implies(implies(a, b), implies(implies(b, c), implies(a, c))))   cnf(prove_cn$_{1}$, negated_conjecture)

LCL064-2.p CN-1 depends on the Church system
Axiomatisations of the Implication/Negation 2 valued sentential calculus are CN-1,CN-2,CN-3 by Lukasiewicz, CN-18,CN-21,CN-35,CN-39,CN-39,CN-40,CN-46 by Frege, CN-3,CN-18,CN-21,CN-22,CN-30,CN-54 by Hilbert, CN-18, CN-35,CN-49 by Church, CN-19,CN-37,CN-59 by Lukasiewicz, CN-19,CN-37,CN-60 by Wos, and the single Meredith axiom. Show that CN-1 depends on a simplified Church system.

(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(x, implies(y, x))) cnf(cn_{18}, axiom)
is_a_theorem(implies(implies(x, implies(y, z)), implies(implies(x, y), implies(x, z)))) cnf(cn_{35}, axiom)
$\neg$ is_a_theorem(implies(implies(a, b), implies(implies(b, c), implies(a, c)))) cnf(prove_cn_1, negated_conjecture)

LCL065-1.p CN-2 depends on the Church system
Axiomatisations of the Implication/Negation 2 valued sentential calculus are CN-1,CN-2,CN-3 by Lukasiewicz, CN-18,CN-21,CN-35,CN-39,CN-39,CN-40,CN-46 by Frege, CN-3,CN-18,CN-21,CN-22,CN-30,CN-54 by Hilbert, CN-18, CN-35,CN-49 by Church, CN-19,CN-37,CN-59 by Lukasiewicz, CN-19,CN-37,CN-60 by Wos, and the single Meredith axiom. Show that CN-2 depends on the Church system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(x, implies(y, x))) cnf(cn_{18}, axiom)
is_a_theorem(implies(implies(x, implies(y, z)), implies(implies(x, y), implies(x, z)))) cnf(cn_{35}, axiom)
is_a_theorem(implies(implies(not(x), not(y)), implies(y, x))) cnf(cn_{49}, axiom)
$\neg$ is_a_theorem(implies(implies(not(a), a), a)) cnf(prove_cn_2, negated_conjecture)

LCL066-1.p CN-3 depends on the Church system
Axiomatisations of the Implication/Negation 2 valued sentential calculus are CN-1,CN-2,CN-3 by Lukasiewicz, CN-18,CN-21,CN-35,CN-39,CN-39,CN-40,CN-46 by Frege, CN-3,CN-18,CN-21,CN-22,CN-30,CN-54 by Hilbert, CN-18, CN-35,CN-49 by Church, CN-19,CN-37,CN-59 by Lukasiewicz, CN-19,CN-37,CN-60 by Wos, and the single Meredith axiom. Show that CN-3 depends on the Church system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(x, implies(y, x))) cnf(cn_{18}, axiom)
is_a_theorem(implies(implies(x, implies(y, z)), implies(implies(x, y), implies(x, z)))) cnf(cn_{35}, axiom)
is_a_theorem(implies(implies(not(x), not(y)), implies(y, x))) cnf(cn_{49}, axiom)
$\neg$ is_a_theorem(implies(a, implies(not(a), b))) cnf(prove_cn_3, negated_conjecture)

LCL067-1.p CN-1 depends on the second Lukasiewicz system
Axiomatisations of the Implication/Negation 2 valued sentential calculus are CN-1,CN-2,CN-3 by Lukasiewicz, CN-18,CN-21,CN-35,CN-39,CN-39,CN-40,CN-46 by Frege, CN-3,CN-18,CN-21,CN-22,CN-30,CN-54 by Hilbert, CN-18, CN-35,CN-49 by Church, CN-19,CN-37,CN-59 by Lukasiewicz, CN-19,CN-37,CN-60 by Wos, and the single Meredith axiom. Show that CN-1 depends on the second Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(implies(x, y), z), implies(y, z))) cnf(cn_{19}, axiom)
is_a_theorem(implies(implies(implies(x, y), z), implies(not(x), z))) cnf(cn_{37}, axiom)
is_a_theorem(implies(implies(not(x), z), implies(implies(y, z), implies(implies(x, y), z)))) cnf(cn_{59}, axiom)
$\neg$ is_a_theorem(implies(implies(a, b), implies(implies(b, c), implies(a, c)))) cnf(prove_cn_1, negated_conjecture)

LCL068-1.p CN-2 depends on the second Lukasiewicz system
Axiomatisations of the Implication/Negation 2 valued sentential calculus are CN-1,CN-2,CN-3 by Lukasiewicz, CN-18,CN-21,CN-35,CN-39,CN-39,CN-40,CN-46 by Frege, CN-3,CN-18,CN-21,CN-22,CN-30,CN-54 by Hilbert, CN-18, CN-35,CN-49 by Church, CN-19,CN-37,CN-59 by Lukasiewicz, CN-19,CN-37,CN-60 by Wos, and the single Meredith axiom. Show that CN-2 depends on the second Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(implies(x, y), z), implies(y, z))) cnf(cn_{19}, axiom)
is_a_theorem(implies(implies(implies(x, y), z), implies(not(x), z))) cnf(cn_{37}, axiom)
is_a_theorem(implies(implies(not(x), z), implies(implies(y, z), implies(implies(x, y), z)))) cnf(cn_{59}, axiom)
$\neg$ is_a_theorem(implies(implies(not(a), a), a)) cnf(prove_cn_2, negated_conjecture)

LCL069-1.p CN-3 depends on the second Lukasiewicz system
Axiomatisations of the Implication/Negation 2 valued sentential calculus are CN-1,CN-2,CN-3 by Lukasiewicz, CN-18,CN-21,CN-35,CN-39,CN-39,CN-40,CN-46 by Frege, CN-3,CN-18,CN-21,CN-22,CN-30,CN-54 by Hilbert, CN-18, CN-35,CN-49 by Church, CN-19,CN-37,CN-59 by Lukasiewicz, CN-19,CN-37,CN-60 by Wos, and the single Meredith axiom. Show that CN-3 depends on the second Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(implies(x, y), z), implies(y, z))) cnf(cn_{19}, axiom)
is_a_theorem(implies(implies(implies(x, y), z), implies(not(x), z))) cnf(cn_{37}, axiom)
is_a_theorem(implies(implies(not(x), z), implies(implies(y, z), implies(implies(x, y), z)))) cnf(cn_{59}, axiom)
$\neg$ is_a_theorem(implies(a, implies(not(a), b))) cnf(prove_cn_3, negated_conjecture)

LCL070-1.p CN-1 depends on the Wos system
Axiomatisations of the Implication/Negation 2 valued sentential calculus are CN-1,CN-2,CN-3 by Lukasiewicz, CN-18,CN-21,CN-35,CN-39,CN-39,CN-40,CN-46 by Frege, CN-3,CN-18,CN-21,CN-22,CN-30,CN-54 by Hilbert, CN-18,

CN-35,CN-49 by Church, CN-19,CN-37,CN-59 by Lukasiewicz, CN-19,CN-37,CN-60 by Wos, and the single Meredith axiom. Show that CN-1 depends on the Wos system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(implies(x, y), z), implies(y, z))) cnf(cn$_{19}$, axiom)
is_a_theorem(implies(implies(implies(x, y), z), implies(not(x), z))) cnf(cn$_{37}$, axiom)
is_a_theorem(implies(implies(x, implies(not(y), z)), implies(x, implies(implies(u, z), implies(implies(y, u), z))))) cnf(cn$_{60}$, ax
$\neg$ is_a_theorem(implies(implies(a, b), implies(implies(b, c), implies(a, c)))) cnf(prove_cn$_1$, negated_conjecture)

LCL071-1.p CN-2 depends on the Wos system
Axiomatisations of the Implication/Negation 2 valued sentential calculus are CN-1,CN-2,CN-3 by Lukasiewicz, CN-18,CN-21,CN-35,CN-39,CN-39,CN-40,CN-46 by Frege, CN-3,CN-18,CN-21,CN-22,CN-30,CN-54 by Hilbert, CN-18, CN-35,CN-49 by Church, CN-19,CN-37,CN-59 by Lukasiewicz, CN-19,CN-37,CN-60 by Wos, and the single Meredith axiom. Show that CN-2 depends on the Wos system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(implies(x, y), z), implies(y, z))) cnf(cn$_{19}$, axiom)
is_a_theorem(implies(implies(implies(x, y), z), implies(not(x), z))) cnf(cn$_{37}$, axiom)
is_a_theorem(implies(implies(x, implies(not(y), z)), implies(x, implies(implies(u, z), implies(implies(y, u), z))))) cnf(cn$_{60}$, ax
$\neg$ is_a_theorem(implies(implies(not(a), a), a)) cnf(prove_cn$_2$, negated_conjecture)

LCL072-1.p CN-3 depends on the Wos system
Axiomatisations of the Implication/Negation 2 valued sentential calculus are CN-1,CN-2,CN-3 by Lukasiewicz, CN-18,CN-21,CN-35,CN-39,CN-39,CN-40,CN-46 by Frege, CN-3,CN-18,CN-21,CN-22,CN-30,CN-54 by Hilbert, CN-18, CN-35,CN-49 by Church, CN-19,CN-37,CN-59 by Lukasiewicz, CN-19,CN-37,CN-60 by Wos, and the single Meredith axiom. Show that CN-3 depends on the Wos system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(implies(x, y), z), implies(y, z))) cnf(cn$_{19}$, axiom)
is_a_theorem(implies(implies(implies(x, y), z), implies(not(x), z))) cnf(cn$_{37}$, axiom)
is_a_theorem(implies(implies(x, implies(not(y), z)), implies(x, implies(implies(u, z), implies(implies(y, u), z))))) cnf(cn$_{60}$, ax
$\neg$ is_a_theorem(implies(a, implies(not(a), b))) cnf(prove_cn$_3$, negated_conjecture)

LCL073-1.p CN-1 depends on the single Merideth axiom
Axiomatisations of the Implication/Negation 2 valued sentential calculus are CN-1,CN-2,CN-3 by Lukasiewicz, CN-18,CN-21,CN-35,CN-39,CN-39,CN-40,CN-46 by Frege, CN-3,CN-18,CN-21,CN-22,CN-30,CN-54 by Hilbert, CN-18, CN-35,CN-49 by Church, CN-19,CN-37,CN-59 by Lukasiewicz, CN-19,CN-37,CN-60 by Wos, and the single Meredith axiom. Show that CN-1 depends on the single Meredith axiom.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(implies(implies(implies(x, y), implies(not(z), not(u))), z), v), implies(implies(v, x), implies(u, x)))
$\neg$ is_a_theorem(implies(implies(a, b), implies(implies(b, c), implies(a, c)))) cnf(prove_cn$_1$, negated_conjecture)

LCL074-1.p CN-2 depends on the single Merideth axiom
Axiomatisations of the Implication/Negation 2 valued sentential calculus are CN-1,CN-2,CN-3 by Lukasiewicz, CN-18,CN-21,CN-35,CN-39,CN-39,CN-40,CN-46 by Frege, CN-3,CN-18,CN-21,CN-22,CN-30,CN-54 by Hilbert, CN-18, CN-35,CN-49 by Church, CN-19,CN-37,CN-59 by Lukasiewicz, CN-19,CN-37,CN-60 by Wos, and the single Meredith axiom. Show that CN-2 depends on the single Meredith axiom.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(implies(implies(implies(x, y), implies(not(z), not(u))), z), v), implies(implies(v, x), implies(u, x)))
$\neg$ is_a_theorem(implies(implies(not(a), a), a)) cnf(prove_cn$_2$, negated_conjecture)

LCL075-1.p CN-3 depends on the single Merideth axiom
Axiomatisations of the Implication/Negation 2 valued sentential calculus are CN-1,CN-2,CN-3 by Lukasiewicz, CN-18,CN-21,CN-35,CN-39,CN-39,CN-40,CN-46 by Frege, CN-3,CN-18,CN-21,CN-22,CN-30,CN-54 by Hilbert, CN-18, CN-35,CN-49 by Church, CN-19,CN-37,CN-59 by Lukasiewicz, CN-19,CN-37,CN-60 by Wos, and the single Meredith axiom. Show that CN-3 depends on the single Meredith axiom.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(implies(implies(implies(x, y), implies(not(z), not(u))), z), v), implies(implies(v, x), implies(u, x)))
$\neg$ is_a_theorem(implies(a, implies(not(a), b))) cnf(prove_cn$_3$, negated_conjecture)

LCL076-1.p CN-40 depends on the Church system
Axiomatisations of the Implication/Negation 2 valued sentential calculus are CN-1,CN-2,CN-3 by Lukasiewicz, CN-18,CN-21,CN-35,CN-39,CN-39,CN-40,CN-46 by Frege, CN-3,CN-18,CN-21,CN-22,CN-30,CN-54 by Hilbert, CN-18, CN-35,CN-49 by Church, CN-19,CN-37,CN-59 by Lukasiewicz, CN-19,CN-37,CN-60 by Wos, and the single Meredith axiom. Show that CN-40 depends on the Church system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)

is_a_theorem(implies(x, implies(y, x))) cnf(cn$_{18}$, axiom)
is_a_theorem(implies(implies(x, implies(y, z)), implies(implies(x, y), implies(x, z)))) cnf(cn$_{35}$, axiom)
is_a_theorem(implies(implies(not(x), not(y)), implies(y, x))) cnf(cn$_{49}$, axiom)
$\neg$ is_a_theorem(implies(a, not(not(a)))) cnf(prove_cn$_{40}$, negated_conjecture)

LCL076-2.p CN-40 depends on the Church system
Axiomatisations of the Implication/Negation 2 valued sentential calculus are CN-1,CN-2,CN-3 by Lukasiewicz, CN-18,CN-21,CN-35,CN-39,CN-39,CN-40,CN-46 by Frege, CN-3,CN-18,CN-21,CN-22,CN-30,CN-54 by Hilbert, CN-18, CN-35,CN-49 by Church, CN-19,CN-37,CN-59 by Lukasiewicz, CN-19,CN-37,CN-60 by Wos, and the single Meredith axiom. Show that CN-40 depends on the Church system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(x, implies(y, x))) cnf(cn$_{18}$, axiom)
is_a_theorem(implies(implies(x, implies(y, z)), implies(implies(x, y), implies(x, z)))) cnf(cn$_{35}$, axiom)
is_a_theorem(implies(implies(not(x), not(y)), implies(y, x))) cnf(cn$_{49}$, axiom)
is_a_theorem(implies(not(not(x_1)), x_1)) cnf(extra_lemma, axiom)
$\neg$ is_a_theorem(implies(a, not(not(a)))) cnf(prove_cn$_{40}$, negated_conjecture)

LCL076-3.p CN-40 depends on the Church system
Axiomatisations of the Implication/Negation 2 valued sentential calculus are CN-1,CN-2,CN-3 by Lukasiewicz, CN-18,CN-21,CN-35,CN-39,CN-39,CN-40,CN-46 by Frege, CN-3,CN-18,CN-21,CN-22,CN-30,CN-54 by Hilbert, CN-18, CN-35,CN-49 by Church, CN-19,CN-37,CN-59 by Lukasiewicz, CN-19,CN-37,CN-60 by Wos, and the single Meredith axiom. Show that CN-40 depends on the Church system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(x, implies(y, x))) cnf(cn$_{18}$, axiom)
is_a_theorem(implies(implies(x, implies(y, z)), implies(implies(x, y), implies(x, z)))) cnf(cn$_{35}$, axiom)
is_a_theorem(implies(implies(not(x), not(y)), implies(y, x))) cnf(cn$_{49}$, axiom)
(is_a_theorem(implies(x_1, x_2)) and is_a_theorem(implies(x_2, x_3))) $\Rightarrow$ is_a_theorem(implies(x_1, x_3)) cnf(transitivity, axiom
$\neg$ is_a_theorem(implies(a, not(not(a)))) cnf(prove_cn$_{40}$, negated_conjecture)

LCL077-1.p CN-39 depends on the Church system
Axiomatisations of the Implication/Negation 2 valued sentential calculus are CN-1,CN-2,CN-3 by Lukasiewicz, CN-18,CN-21,CN-35,CN-39,CN-39,CN-40,CN-46 by Frege, CN-3,CN-18,CN-21,CN-22,CN-30,CN-54 by Hilbert, CN-18, CN-35,CN-49 by Church, CN-19,CN-37,CN-59 by Lukasiewicz, CN-19,CN-37,CN-60 by Wos, and the single Meredith axiom. Show that CN-39 depends on the Church system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(x, implies(y, x))) cnf(cn$_{18}$, axiom)
is_a_theorem(implies(implies(x, implies(y, z)), implies(implies(x, y), implies(x, z)))) cnf(cn$_{35}$, axiom)
is_a_theorem(implies(implies(not(x), not(y)), implies(y, x))) cnf(cn$_{49}$, axiom)
$\neg$ is_a_theorem(implies(not(not(a)), a)) cnf(prove_cn$_{39}$, negated_conjecture)

LCL077-2.p CN-39 depends on the Church system
Axiomatisations of the Implication/Negation 2 valued sentential calculus are CN-1,CN-2,CN-3 by Lukasiewicz, CN-18,CN-21,CN-35,CN-39,CN-39,CN-40,CN-46 by Frege, CN-3,CN-18,CN-21,CN-22,CN-30,CN-54 by Hilbert, CN-18, CN-35,CN-49 by Church, CN-19,CN-37,CN-59 by Lukasiewicz, CN-19,CN-37,CN-60 by Wos, and the single Meredith axiom. Show that CN-39 depends on the Church system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(x, implies(y, x))) cnf(cn$_{18}$, axiom)
is_a_theorem(implies(implies(x, implies(y, z)), implies(implies(x, y), implies(x, z)))) cnf(cn$_{35}$, axiom)
is_a_theorem(implies(implies(not(x), not(y)), implies(y, x))) cnf(cn$_{49}$, axiom)
(is_a_theorem(implies(x_1, x_2)) and is_a_theorem(implies(x_2, x_3))) $\Rightarrow$ is_a_theorem(implies(x_1, x_3)) cnf(transitivity, axiom
$\neg$ is_a_theorem(implies(not(not(a)), a)) cnf(prove_cn$_{39}$, negated_conjecture)

LCL078-1.p CN-40 depends on CN-18 CN-35 CN-46
Axiomatisations of the Implication/Negation 2 valued sentential calculus are CN-1,CN-2,CN-3 by Lukasiewicz, CN-18,CN-21,CN-35,CN-39,CN-39,CN-40,CN-46 by Frege, CN-3,CN-18,CN-21,CN-22,CN-30,CN-54 by Hilbert, CN-18, CN-35,CN-49 by Church, CN-19,CN-37,CN-59 by Lukasiewicz, CN-19,CN-37,CN-60 by Wos, and the single Meredith axiom. Show that CN-40 depends on the modified Church system CN-18,CN-35,CN-46.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(x, implies(y, x))) cnf(cn$_{18}$, axiom)
is_a_theorem(implies(implies(x, implies(y, z)), implies(implies(x, y), implies(x, z)))) cnf(cn$_{35}$, axiom)
is_a_theorem(implies(implies(y, x), implies(not(x), not(y)))) cnf(cn$_{46}$, axiom)
$\neg$ is_a_theorem(implies(a, not(not(a)))) cnf(prove_cn$_{40}$, negated_conjecture)

LCL079-1.p Transitivity can be derived from Church's system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-18,CN-35,CN-49 by Church. Show that transitivity of implies can be derived from the Church system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(implies(x, implies(y, x)))   cnf(cn$_{18}$, axiom)
is_a_theorem(implies(implies(x, implies(y, z)), implies(implies(x, y), implies(x, z))))   cnf(cn$_{35}$, axiom)
is_a_theorem(implies(implies(y, x), implies(not(x), not(y))))   cnf(cn_49_reversed, axiom)
is_a_theorem(implies(a, b))   cnf(a_implies_b, hypothesis)
is_a_theorem(implies(b, c))   cnf(b_implies_c, hypothesis)
$\neg$ is_a_theorem(implies(a, c))   cnf(prove_transitivity, negated_conjecture)

LCL080-1.p The 1st Lukasiewicz axiom depends on Tarski-Bernays system
Axiomatisations of the Implicational propositional calculus are IC-2,IC-3,IC-4 by Tarski-Bernays and single Lukasiewicz axioms.Show that the 1st Lukasiewicz axiom depends on the Tarski-Bernays system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(implies(x, implies(y, x)))   cnf(ic$_2$, axiom)
is_a_theorem(implies(implies(implies(x, y), x), x))   cnf(ic$_3$, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z))))   cnf(ic$_4$, axiom)
$\neg$ is_a_theorem(implies(implies(implies(a, b), c), implies(implies(c, a), implies(e, a))))   cnf(prove_ic_JLukasiewicz, negated_co

LCL080-2.p The 1st Lukasiewicz axiom depends on Tarski-Bernays system
Axiomatisations of the Implicational propositional calculus are IC-2,IC-3,IC-4 by Tarski-Bernays and Lukasiewicz axioms. Show that the 1st Lukasiewicz axiom depends on the Tarski-Bernays system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(implies(x, x))   cnf(ic$_1$, axiom)
is_a_theorem(implies(x, implies(y, x)))   cnf(ic$_2$, axiom)
is_a_theorem(implies(implies(implies(x, y), x), x))   cnf(ic$_3$, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z))))   cnf(ic$_4$, axiom)
$\neg$ is_a_theorem(implies(implies(implies(a, b), c), implies(implies(c, a), implies(e, a))))   cnf(prove_ic_JLukasiewicz, negated_co

LCL081-1.p IC-1 depends on the 1st Lukasiewicz axiom
Axiomatisations of the Implicational propositional calculus are IC-2,IC-3,IC-4 by Tarski-Bernays and single Lukasiewicz axioms.Show that IC-1 depends on the first Lukasiewicz axiom.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(implies(x, y), z), implies(implies(z, x), implies(u, x))))   cnf(ic_JLukasiewicz, axiom)
$\neg$ is_a_theorem(implies(a, a))   cnf(prove_ic$_1$, negated_conjecture)

LCL082-1.p IC-2 depends on the 1st Lukasiewicz axiom
Axiomatisations of the Implicational propositional calculus are IC-2,IC-3,IC-4 by Tarski-Bernays and single Lukasiewicz axioms.Show that IC-2 depends on the first Lukasiewicz axiom.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(implies(x, y), z), implies(implies(z, x), implies(u, x))))   cnf(ic_JLukasiewicz, axiom)
$\neg$ is_a_theorem(implies(a, implies(b, a)))   cnf(prove_ic$_2$, negated_conjecture)

LCL083-1.p IC-3 depends on the 1st Lukasiewicz axiom
Axiomatisations of the Implicational propositional calculus are IC-2,IC-3,IC-4 by Tarski-Bernays and single Lukasiewicz axioms. Show that IC-3 depends on the first Lukasiewicz axiom.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(implies(x, y), z), implies(implies(z, x), implies(u, x))))   cnf(ic_JLukasiewicz, axiom)
$\neg$ is_a_theorem(implies(implies(implies(a, b), a), a))   cnf(prove_ic$_3$, negated_conjecture)

LCL083-2.p IC-3 depends on the 1st Lukasiewicz axiom
Axiomatisations of the Implicational propositional calculus are IC-2,IC-3,IC-4 by Tarski-Bernays and single Lukasiewicz axioms. Show that IC-3 depends on the first Lukasiewicz axiom.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(implies(x, y), z), implies(implies(z, x), implies(u, x))))   cnf(ic_JLukasiewicz, axiom)
is_a_theorem(implies(x, x))   cnf(ic$_1$, axiom)
$\neg$ is_a_theorem(implies(implies(implies(a, b), a), a))   cnf(prove_ic$_3$, negated_conjecture)

LCL084-2.p IC-4 depends on the 1st Lukasiewicz axiom
Axiomatisations of the Implicational propositional calculus are IC-2,IC-3,IC-4 by Tarski-Bernays and single Lukasiewicz axioms. Show that IC-4 depends on the first Lukasiewicz axiom.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y)   cnf(condensed_detachment, axiom)

is_a_theorem(implies(implies(implies(x, y), z), implies(implies(z, x), implies(u, x)))) cnf(ic_JLukasiewicz, axiom)
is_a_theorem(implies(x, x)) cnf(ic$_1$, axiom)
¬ is_a_theorem(implies(implies(a, b), implies(implies(b, c), implies(a, c)))) cnf(prove_ic$_4$, negated_conjecture)

LCL084-3.p IC-4 depends on the 1st Lukasiewicz axiom
Axiomatisations of the Implicational propositional calculus are IC-2,IC-3,IC-4 by Tarski-Bernays and single Lukasiewicz axioms. Show that IC-4 depends on the first Lukasiewicz axiom.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(implies(x, y), z), implies(implies(z, x), implies(u, x)))) cnf(ic_JLukasiewicz, axiom)
is_a_theorem(implies(x, x)) cnf(ic$_1$, axiom)
is_a_theorem(implies(implies(implies(x, y), x), x)) cnf(ic$_3$, axiom)
¬ is_a_theorem(implies(implies(a, b), implies(implies(b, c), implies(a, c)))) cnf(prove_ic$_4$, negated_conjecture)

LCL085-1.p IC-5 depends on the 1st Lukasiewicz axiom
Axiomatisations of the Implicational propositional calculus are IC-2,IC-3,IC-4 by Tarski-Bernays and single Lukasiewicz axioms. Show that IC-5 depends on the first Lukasiewicz axiom.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(implies(x, y), z), implies(implies(z, x), implies(u, x)))) cnf(ic_JLukasiewicz, axiom)
¬ is_a_theorem(implies(a, implies(implies(a, b), b))) cnf(prove_ic$_5$, negated_conjecture)

LCL086-1.p IC-1 depends on the 4th Lukasiewicz axiom
Axiomatisations of the Implicational propositional calculus are IC-2,IC-3,IC-4 by Tarski-Bernays and single Lukasiewicz axioms. Show that IC-1 depends on the fourth Lukasiewicz axiom.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(implies(p, q), implies(r, s)), implies(t, implies(implies(s, p), implies(r, p))))) cnf(ic_JLukasiewi
¬ is_a_theorem(implies(a, a)) cnf(prove_ic$_1$, negated_conjecture)

LCL087-1.p IC-2 depends on the 4th Lukasiewicz axiom
Axiomatisations of the Implicational propositional calculus are IC-2,IC-3,IC-4 by Tarski-Bernays and single Lukasiewicz axioms. Show that IC-2 depends on the fourth Lukasiewicz axiom.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(implies(p, q), implies(r, s)), implies(t, implies(implies(s, p), implies(r, p))))) cnf(ic_JLukasiewic
¬ is_a_theorem(implies(a, implies(b, a))) cnf(prove_ic$_2$, negated_conjecture)

LCL088-1.p IC-3 depends on the 4th Lukasiewicz axiom
Axiomatisations of the Implicational propositional calculus are IC-2,IC-3,IC-4 by Tarski-Bernays and single Lukasiewicz axioms.Show that IC-3 depends on the fourth Lukasiewicz axiom.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(implies(p, q), implies(r, s)), implies(t, implies(implies(s, p), implies(r, p))))) cnf(ic_JLukasiewic
¬ is_a_theorem(implies(implies(implies(a, b), a), a)) cnf(prove_ic$_3$, negated_conjecture)

LCL089-1.p IC-4 depends on the 4th Lukasiewicz axiom
Axiomatisations of the Implicational propositional calculus are IC-2,IC-3,IC-4 by Tarski-Bernays and single Lukasiewicz axioms.Show that IC-4 depends on the fourth Lukasiewicz axiom.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(implies(p, q), implies(r, s)), implies(t, implies(implies(s, p), implies(r, p))))) cnf(ic_JLukasiewic
¬ is_a_theorem(implies(implies(a, b), implies(implies(b, c), implies(a, c)))) cnf(prove_ic$_4$, negated_conjecture)

LCL090-1.p IC-1 depends on the 5th Lukasiewicz axiom
Axiomatisations of the Implicational propositional calculus are IC-2,IC-3,IC-4 by Tarski-Bernays and single Lukasiewicz axioms.Show that IC-1 depends on the fifth Lukasiewicz axiom.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(implies(p, q), implies(r, s)), implies(implies(s, p), implies(t, implies(r, p))))) cnf(ic_JLukasiewic
¬ is_a_theorem(implies(a, a)) cnf(prove_ic$_1$, negated_conjecture)

LCL091-1.p IC-2 depends on the 5th Lukasiewicz axiom
Axiomatisations of the Implicational propositional calculus are IC-2,IC-3,IC-4 by Tarski-Bernays and single Lukasiewicz axioms.Show that IC-2 depends on the fifth Lukasiewicz axiom.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(implies(p, q), implies(r, s)), implies(implies(s, p), implies(t, implies(r, p))))) cnf(ic_JLukasiewic
¬ is_a_theorem(implies(a, implies(b, a))) cnf(prove_ic$_2$, negated_conjecture)

LCL092-1.p IC-3 depends on the 5th Lukasiewicz axiom
Axiomatisations of the Implicational propositional calculus are IC-2,IC-3,IC-4 by Tarski-Bernays and single Lukasiewicz axioms.Show that IC-3 depends on the fifth Lukasiewicz axiom.

(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(implies(p, q), implies(r, s)), implies(implies(s, p), implies(t, implies(r, p))))) cnf(ic_JLukasiewi
$\neg$ is_a_theorem(implies(implies(implies(a, b), a), a)) cnf(prove_ic$_3$, negated_conjecture)

LCL093-1.p IC-4 depends on the 5th Lukasiewicz axiom
Axiomatisations of the Implicational propositional calculus are IC-2,IC-3,IC-4 by Tarski-Bernays and single Lukasiewicz axioms.Show that IC-4 depends on the fifth Lukasiewicz axiom.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(implies(p, q), implies(r, s)), implies(implies(s, p), implies(t, implies(r, p))))) cnf(ic_JLukasiewi
$\neg$ is_a_theorem(implies(implies(a, b), implies(implies(b, c), implies(a, c)))) cnf(prove_ic$_4$, negated_conjecture)

LCL094-1.p IC-5 depends on the 4th Lukasiewicz axiom
Axiomatisations of the Implicational propositional calculus are IC-2,IC-3,IC-4 by Tarski-Bernays and single Lukasiewicz axioms.Show that IC-5 depends on the fourth Lukasiewicz axiom.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(implies(p, q), implies(r, s)), implies(t, implies(implies(s, p), implies(r, p))))) cnf(ic_JLukasiewi
$\neg$ is_a_theorem(implies(a, implies(implies(a, b), b))) cnf(prove_ic$_5$, negated_conjecture)

LCL095-1.p IC-5 depends on the 5th Lukasiewicz axiom
Axiomatisations of the Implicational propositional calculus are IC-2,IC-3,IC-4 by Tarski-Bernays and single Lukasiewicz axioms.Show that IC-5 depends on the fifth Lukasiewicz axiom.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(implies(p, q), implies(r, s)), implies(implies(s, p), implies(t, implies(r, p))))) cnf(ic_JLukasiewi
$\neg$ is_a_theorem(implies(a, implies(implies(a, b), b))) cnf(prove_ic$_5$, negated_conjecture)

LCL096-1.p LG-1 depends on LG-2, LG-3, LG-4
Axiomatisations of the left group calculus are LG-1, LG-2,LG-3,LG-4,LG-5 by Kalman, LG-2,LG-3, LG-2,P-1, LG-2,P-4, LG-2,Q-1,Q-2, P-1,Q-3, P-4,Q-3, Q-1, Q-2,Q-3, Q-1,Q-3,Q-4, LG-27-1690 all by McCune. Show that LG-1 depends on a part of the Kalman system.
(is_a_theorem(equivalent(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(equivalent(equivalent(equivalent(equivalent(equivalent(x, y), equivalent(x, z)), equivalent(y, z)), u), u)) cnf(lg
is_a_theorem(equivalent(equivalent(equivalent(equivalent(equivalent(equivalent(x, y), equivalent(x, z)), u), equivalent(equivale
is_a_theorem(equivalent(equivalent(equivalent(equivalent(x, y), z), u), equivalent(equivalent(equivalent(x, v), z), equivalent(eq
$\neg$ is_a_theorem(equivalent(equivalent(equivalent(a, equivalent(equivalent(b, b), a)), c), c)) cnf(prove_lg$_1$, negated_conjecture

LCL097-1.p LG-4 depends on LG-2, LG-3
Axiomatisations of the left group calculus are LG-1, LG-2,LG-3,LG-4,LG-5 by Kalman, LG-2,LG-3, LG-2,P-1, LG-2,P-4, LG-2,Q-1,Q-2, P-1,Q-3, P-4,Q-3, Q-1, Q-2,Q-3, Q-1,Q-3,Q-4, LG-27-1690 all by McCune. Show that LG-4 depends on a part of the Kalman system.
(is_a_theorem(equivalent(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(equivalent(equivalent(equivalent(equivalent(equivalent(x, y), equivalent(x, z)), equivalent(y, z)), u), u)) cnf(lg
is_a_theorem(equivalent(equivalent(equivalent(equivalent(equivalent(equivalent(x, y), equivalent(x, z)), u), equivalent(equivale
$\neg$ is_a_theorem(equivalent(equivalent(equivalent(equivalent(a, b), c), e), equivalent(equivalent(equivalent(a, falsehood), c), equi

LCL098-1.p LG-4 depends on LG-3
Axiomatisations of the left group calculus are LG-1, LG-2,LG-3,LG-4,LG-5 by Kalman, LG-2,LG-3, LG-2,P-1, LG-2,P-4, LG-2,Q-1,Q-2, P-1,Q-3, P-4,Q-3, Q-1, Q-2,Q-3, Q-1,Q-3,Q-4, LG-27-1690 all by McCune. Show that LG-4 depends on LG-3.
(is_a_theorem(equivalent(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(equivalent(equivalent(equivalent(equivalent(equivalent(equivalent(x, y), equivalent(x, z)), u), equivalent(equivale
$\neg$ is_a_theorem(equivalent(equivalent(equivalent(equivalent(a, b), c), e), equivalent(equivalent(equivalent(a, falsehood), c), equi

LCL099-1.p LG-5 depends on the 1st McCune system
Axiomatisations of the left group calculus are LG-1, LG-2,LG-3,LG-4,LG-5 by Kalman, LG-2,LG-3, LG-2,P-1, LG-2,P-4, LG-2,Q-1,Q-2, P-1,Q-3, P-4,Q-3, Q-1, Q-2,Q-3, Q-1,Q-3,Q-4, LG-27-1690 all by McCune. Show that LG-5 depends on the first McCune system.
(is_a_theorem(equivalent(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(equivalent(equivalent(equivalent(equivalent(equivalent(x, y), equivalent(x, z)), equivalent(y, z)), u), u)) cnf(lg
is_a_theorem(equivalent(equivalent(equivalent(equivalent(equivalent(equivalent(x, y), equivalent(x, z)), u), equivalent(equivale
$\neg$ is_a_theorem(equivalent(equivalent(equivalent(a, equivalent(equivalent(b, a), c)), equivalent(equivalent(e, a), falsehood)), equ

LCL100-1.p LG-3 depends on the 2nd McCune system

Axiomatisations of the left group calculus are LG-1, LG-2,LG-3,LG-4,LG-5 by Kalman, LG-2,LG-3, LG-2,P-1, LG-2,P-4, LG-2,Q-1,Q-2, P-1,Q-3, P-4,Q-3, Q-1, Q-2,Q-3, Q-1,Q-3,Q-4, LG-27-1690 all by McCune. Show that LG-3 depends on the second McCune system.

(is_a_theorem(equivalent(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(equivalent(equivalent(equivalent(equivalent(equivalent(x, y), equivalent(x, z)), equivalent(y, z)), u), u))   cnf(lg
is_a_theorem(equivalent(equivalent(equivalent(x, y), z), equivalent(equivalent(u, y), equivalent(equivalent(x, u), z))))   cnf(p
¬ is_a_theorem(equivalent(equivalent(equivalent(equivalent(equivalent(equivalent(a, b), equivalent(a, c)), e), equival

LCL101-1.p P-1 depends on the 3rd McCune system

Axiomatisations of the left group calculus are LG-1, LG-2,LG-3,LG-4,LG-5 by Kalman, LG-2,LG-3, LG-2,P-1, LG-2,P-4, LG-2,Q-1,Q-2, P-1,Q-3, P-4,Q-3, Q-1, Q-2,Q-3, Q-1,Q-3,Q-4, LG-27-1690 all by McCune. Show that P-1 depends on the 3rd McCune system.

(is_a_theorem(equivalent(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(equivalent(equivalent(equivalent(equivalent(equivalent(x, y), equivalent(x, z)), equivalent(y, z)), u), u))   cnf(lg
is_a_theorem(equivalent(x, equivalent(equivalent(equivalent(equivalent(y, z), equivalent(y, u)), equivalent(z, u)), x)))   cnf(p
¬ is_a_theorem(equivalent(equivalent(equivalent(a, b), c), equivalent(equivalent(e, b), equivalent(equivalent(a, e), c))))   cnf(p

LCL102-1.p P-1 depends on the 4th McCune system

Axiomatisations of the left group calculus are LG-1, LG-2,LG-3,LG-4,LG-5 by Kalman, LG-2,LG-3, LG-2,P-1, LG-2,P-4, LG-2,Q-1,Q-2, P-1,Q-3, P-4,Q-3, Q-1, Q-2,Q-3, Q-1,Q-3,Q-4, LG-27-1690 all by McCune. Show that P-1 depends on the 4th McCune system.

(is_a_theorem(equivalent(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(equivalent(equivalent(equivalent(equivalent(equivalent(x, y), equivalent(x, z)), equivalent(y, z)), u), u))   cnf(lg
is_a_theorem(equivalent(x, equivalent(equivalent(y, z), equivalent(equivalent(z, y), x))))   cnf(q_1, axiom)
is_a_theorem(equivalent(equivalent(x, y), equivalent(equivalent(z, x), equivalent(z, y))))   cnf(q_2, axiom)
¬ is_a_theorem(equivalent(equivalent(equivalent(a, b), c), equivalent(equivalent(e, b), equivalent(equivalent(a, e), c))))   cnf(p

LCL103-1.p LG-2 depends on the 5th McCune system

Axiomatisations of the left group calculus are LG-1, LG-2,LG-3,LG-4,LG-5 by Kalman, LG-2,LG-3, LG-2,P-1, LG-2,P-4, LG-2,Q-1,Q-2, P-1,Q-3, P-4,Q-3, Q-1, Q-2,Q-3, Q-1,Q-3,Q-4, LG-27-1690 all by McCune. Show that LG-2 depends on the 5th McCune system.

(is_a_theorem(equivalent(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(equivalent(equivalent(equivalent(x, y), z), equivalent(equivalent(u, y), equivalent(equivalent(x, u), z))))   cnf(p
is_a_theorem(equivalent(equivalent(equivalent(x, y), equivalent(equivalent(y, x), z)), z))   cnf(q_3, axiom)
¬ is_a_theorem(equivalent(equivalent(equivalent(equivalent(equivalent(a, b), equivalent(a, c)), equivalent(b, c)), e), e))   cnf(p

LCL104-1.p P-1 depends on the 6th McCune system

Axiomatisations of the left group calculus are LG-1, LG-2,LG-3,LG-4,LG-5 by Kalman, LG-2,LG-3, LG-2,P-1, LG-2,P-4, LG-2,Q-1,Q-2, P-1,Q-3, P-4,Q-3, Q-1, Q-2,Q-3, Q-1,Q-3,Q-4, LG-27-1690 all by McCune. Show that P-1 depends on the sixth McCune system.

(is_a_theorem(equivalent(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(equivalent(x, equivalent(equivalent(equivalent(equivalent(y, z), equivalent(y, u)), equivalent(z, u)), x)))   cnf(p
is_a_theorem(equivalent(equivalent(equivalent(x, y), equivalent(equivalent(y, x), z)), z))   cnf(q_3, axiom)
¬ is_a_theorem(equivalent(equivalent(equivalent(a, b), c), equivalent(equivalent(e, b), equivalent(equivalent(a, e), c))))   cnf(p

LCL105-1.p LG-2 depends on the 7th McCune system

Axiomatisations of the left group calculus are LG-1, LG-2,LG-3,LG-4,LG-5 by Kalman, LG-2,LG-3, LG-2,P-1, LG-2,P-4, LG-2,Q-1,Q-2, P-1,Q-3, P-4,Q-3, Q-1, Q-2,Q-3, Q-1,Q-3,Q-4, LG-27-1690 all by McCune. Show that LG-2 depends on the seventh McCune system.

(is_a_theorem(equivalent(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(equivalent(x, equivalent(equivalent(y, z), equivalent(equivalent(z, y), x))))   cnf(q_1, axiom)
is_a_theorem(equivalent(equivalent(x, y), equivalent(equivalent(z, x), equivalent(z, y))))   cnf(q_2, axiom)
is_a_theorem(equivalent(equivalent(equivalent(x, y), equivalent(equivalent(y, x), z)), z))   cnf(q_3, axiom)
¬ is_a_theorem(equivalent(equivalent(equivalent(equivalent(equivalent(a, b), equivalent(a, c)), equivalent(b, c)), e), e))   cnf(p

LCL106-1.p Q-2 depends on Q-1, Q-4

Axiomatisations of the left group calculus are LG-1, LG-2,LG-3,LG-4,LG-5 by Kalman, LG-2,LG-3, LG-2,P-1, LG-2,P-4, LG-2,Q-1,Q-2, P-1,Q-3, P-4,Q-3, Q-1, Q-2,Q-3, Q-1,Q-3,Q-4, LG-27-1690 all by McCune. Show that Q-2 depends on Q-1 and Q-4.

(is_a_theorem(equivalent(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(equivalent(x, equivalent(equivalent(y, z), equivalent(equivalent(z, y), x))))   cnf(q_1, axiom)
is_a_theorem(equivalent(equivalent(equivalent(x, y), equivalent(x, z)), equivalent(y, z)))   cnf(q_4, axiom)
¬ is_a_theorem(equivalent(equivalent(a, b), equivalent(equivalent(c, a), equivalent(c, b))))   cnf(prove_q_2, negated_conjecture

LCL107-1.p P-1 depends on the single McCune axiom
Axiomatisations of the left group calculus are LG-1, LG-2,LG-3,LG-4,LG-5 by Kalman, LG-2,LG-3, LG-2,P-1, LG-2,P-4, LG-2,Q-1,Q-2, P-1,Q-3, P-4,Q-3, Q-1, Q-2,Q-3, Q-1,Q-3,Q-4, LG-27-1690 all by McCune. Show that P-1 depends on the single McCune axiom.
(is_a_theorem(equivalent(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(equivalent(equivalent(equivalent(equivalent(x, y), z), equivalent(equivalent(u, v), equivalent(equivalent(equivalen
$\neg$ is_a_theorem(equivalent(equivalent(equivalent(a, b), c), equivalent(equivalent(e, b), equivalent(equivalent(a, e), c)))) cnf(p

LCL108-1.p Q-3 depends on the single McCune axiom
Axiomatisations of the left group calculus are LG-1, LG-2,LG-3,LG-4,LG-5 by Kalman, LG-2,LG-3, LG-2,P-1, LG-2,P-4, LG-2,Q-1,Q-2, P-1,Q-3, P-4,Q-3, Q-1, Q-2,Q-3, Q-1,Q-3,Q-4, LG-27-1690 all by McCune. Show that Q-3 depends on the single McCune axiom.
(is_a_theorem(equivalent(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(equivalent(equivalent(equivalent(equivalent(x, y), z), equivalent(equivalent(u, v), equivalent(equivalent(equivalen
$\neg$ is_a_theorem(equivalent(equivalent(equivalent(a, b), equivalent(equivalent(b, a), c)), c)) cnf(prove_q$_3$, negated_conjecture)

LCL109-1.p MV-4 depends on the Merideth system
An axiomatisation of the many valued sentential calculus is MV-1,MV-2,MV-3,MV-5 by Meredith. Show that MV-4 depends on the Meredith system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(x, implies(y, x))) cnf(mv$_1$, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z)))) cnf(mv$_2$, axiom)
is_a_theorem(implies(implies(implies(x, y), y), implies(implies(y, x), x))) cnf(mv$_3$, axiom)
is_a_theorem(implies(implies(not(x), not(y)), implies(y, x))) cnf(mv$_5$, axiom)
$\neg$ is_a_theorem(implies(implies(implies(a, b), implies(b, a)), implies(b, a))) cnf(prove_mv$_4$, negated_conjecture)

LCL109-2.p MV-4 depends on the Merideth system
An axiomatisation of the many valued sentential calculus is MV-1,MV-2,MV-3,MV-5 by Meredith. Wajsberg provided a different axiomatisation. Show that MV-4 depends on the Wajsberg system.
include('Axioms/LCL001-0.ax')
implies(implies(implies(a, b), implies(b, a)), implies(b, a)) $\neq$ truth cnf(prove_wajsberg_mv$_4$, negated_conjecture)

LCL109-3.p MV-4 depends on the Merideth system
An axiomatisation of the many valued sentential calculus is MV-1,MV-2,MV-3,MV-5 by Meredith. Show that MV-4 depends on the Meredith system.
include('Axioms/LCL001-0.ax')
implies(x, x) = truth cnf(lemma$_1$, axiom)
implies(x, y) = implies(y, x) $\Rightarrow$ $x = y$ cnf(lemma$_2$, axiom)
implies(x, truth) = truth cnf(lemma$_3$, axiom)
implies(x, implies(y, x)) = truth cnf(lemma$_4$, axiom)
implies(x, y) = implies(y, z) $\Rightarrow$ implies(x, z) = truth cnf(lemma$_5$, axiom)
implies(implies(x, y), implies(implies(z, x), implies(z, y))) = truth cnf(lemma$_6$, axiom)
implies(x, implies(y, z)) = implies(y, implies(x, z)) cnf(lemma$_7$, axiom)
implies(x, not(truth)) = not(x) cnf(lemma$_8$, axiom)
not(not(x)) = x cnf(lemma$_9$, axiom)
implies(not(x), not(y)) = implies(y, x) cnf(lemma$_{10}$, axiom)
implies(implies(implies(a, b), implies(b, a)), implies(b, a)) $\neq$ truth cnf(prove_wajsberg_mv$_4$, negated_conjecture)

LCL109-4.p A theorem in the lattice structure of Wajsberg algebras
include('Axioms/LCL001-0.ax')
include('Axioms/LCL001-1.ax')
big_V(implies(x, y), implies(y, x)) $\neq$ truth cnf(prove_mv$_4$, negated_conjecture)

LCL109-5.p A theorem in the lattice structure of Wajsberg algebras
include('Axioms/LCL001-0.ax')
include('Axioms/LCL001-1.ax')
ordered(x, y) $\Rightarrow$ ordered(implies(x, z), implies(y, z)) cnf(lemma$_1$, axiom)
ordered(x, y) $\Rightarrow$ ordered(implies(z, x), implies(z, y)) cnf(lemma$_2$, axiom)
ordered(x, implies(y, z)) $\Rightarrow$ ordered(y, implies(x, z)) cnf(lemma_3$_1$, axiom)
ordered(y, implies(x, z)) $\Rightarrow$ ordered(x, implies(y, z)) cnf(lemma_3$_2$, axiom)
not(big_V(x, y)) = big_hat(not(x), not(y)) cnf(lemma$_4$, axiom)
not(big_hat(x, y)) = big_V(not(x), not(y)) cnf(lemma$_5$, axiom)
implies(big_V(x, y), z) = big_hat(implies(x, z), implies(y, z)) cnf(lemma$_6$, axiom)

$\text{implies}(x, \text{big\_hat}(y, z)) = \text{big\_hat}(\text{implies}(x, y), \text{implies}(x, z))$ cnf(lemma$_7$, axiom)
$\text{big\_V}(\text{implies}(x, y), \text{implies}(y, x)) \neq \text{truth}$ cnf(prove_mv$_4$, negated_conjecture)

LCL109-6.p A theorem in the lattice structure of Wajsberg algebras
include('Axioms/LCL002-0.ax')
$\text{xor}(x, y) = \text{xor}(y, x)$ cnf(xor_commutativity, axiom)
$\text{and\_star}(\text{and\_star}(x, y), z) = \text{and\_star}(x, \text{and\_star}(y, z))$ cnf(and_star_associativity, axiom)
$\text{and\_star}(x, y) = \text{and\_star}(y, x)$ cnf(and_star_commutativity, axiom)
$\text{not}(\text{truth}) = \text{falsehood}$ cnf(false_definition, axiom)
$\text{implies}(x, y) = \text{xor}(\text{truth}, \text{and\_star}(x, \text{xor}(\text{truth}, y)))$ cnf(implies_definition, axiom)
$\text{implies}(\text{implies}(\text{implies}(a, b), \text{implies}(b, a)), \text{implies}(b, a)) \neq \text{truth}$ cnf(prove_wajsberg_mv$_4$, negated_conjecture)

LCL110-1.p MV-24 depnds on the Merideth system
An axiomatisation of the many valued sentential calculus is MV-1,MV-2,MV-3,MV-5 by Meredith. Show that MV24 depends on the Meredith system.
$(\text{is\_a\_theorem}(\text{implies}(x, y)) \text{ and } \text{is\_a\_theorem}(x)) \Rightarrow \text{is\_a\_theorem}(y)$ cnf(condensed_detachment, axiom)
$\text{is\_a\_theorem}(\text{implies}(x, \text{implies}(y, x)))$ cnf(mv$_1$, axiom)
$\text{is\_a\_theorem}(\text{implies}(\text{implies}(x, y), \text{implies}(\text{implies}(y, z), \text{implies}(x, z))))$ cnf(mv$_2$, axiom)
$\text{is\_a\_theorem}(\text{implies}(\text{implies}(\text{implies}(x, y), y), \text{implies}(\text{implies}(y, x), x)))$ cnf(mv$_3$, axiom)
$\text{is\_a\_theorem}(\text{implies}(\text{implies}(\text{not}(x), \text{not}(y)), \text{implies}(y, x)))$ cnf(mv$_5$, axiom)
$\neg\,\text{is\_a\_theorem}(\text{implies}(\text{not}(\text{not}(a)), a))$ cnf(prove_mv$_{24}$, negated_conjecture)

LCL110-2.p MV-24 depends on the Meredith system
An axiomatisation of the many valued sentential calculus is MV-1,MV-2,MV-3,MV-5 by Meredith. Wajsberg presented an equality axiomatisation. Show that MV-24 depends on the Wajsberg axiomatisation.
include('Axioms/LCL001-0.ax')
$\text{implies}(\text{not}(\text{not}(x)), x) \neq \text{truth}$ cnf(prove_mv$_{24}$, negated_conjecture)

LCL111-1.p MV-25 depends on the Merideth system
An axiomatisation of the many valued sentential calculus is MV-1,MV-2,MV-3,MV-5 by Meredith. Show that MV-25 depends on the Meredith system.
$(\text{is\_a\_theorem}(\text{implies}(x, y)) \text{ and } \text{is\_a\_theorem}(x)) \Rightarrow \text{is\_a\_theorem}(y)$ cnf(condensed_detachment, axiom)
$\text{is\_a\_theorem}(\text{implies}(x, \text{implies}(y, x)))$ cnf(mv$_1$, axiom)
$\text{is\_a\_theorem}(\text{implies}(\text{implies}(x, y), \text{implies}(\text{implies}(y, z), \text{implies}(x, z))))$ cnf(mv$_2$, axiom)
$\text{is\_a\_theorem}(\text{implies}(\text{implies}(\text{implies}(x, y), y), \text{implies}(\text{implies}(y, x), x)))$ cnf(mv$_3$, axiom)
$\text{is\_a\_theorem}(\text{implies}(\text{implies}(\text{not}(x), \text{not}(y)), \text{implies}(y, x)))$ cnf(mv$_5$, axiom)
$\neg\,\text{is\_a\_theorem}(\text{implies}(\text{implies}(a, b), \text{implies}(\text{implies}(c, a), \text{implies}(c, b))))$ cnf(prove_mv$_{25}$, negated_conjecture)

LCL111-2.p MV-25 depends on the Meredith system
An axiomatisation of the many valued sentential calculus is MV-1,MV-2,MV-3,MV-5 by Meredith. Wajsberg presented an equality axiomatisation. Show that MV-25 depends on the Wajsberg axiomatisation.
include('Axioms/LCL001-0.ax')
$\text{implies}(\text{implies}(x, y), \text{implies}(\text{implies}(z, x), \text{implies}(z, y))) \neq \text{truth}$ cnf(prove_mv$_{25}$, negated_conjecture)

LCL112-1.p MV-29 depnds on the Merideth system
An axiomatisation of the many valued sentential calculus is MV-1,MV-2,MV-3,MV-5 by Meredith. Show that 29 depends on the Meredith system.
$(\text{is\_a\_theorem}(\text{implies}(x, y)) \text{ and } \text{is\_a\_theorem}(x)) \Rightarrow \text{is\_a\_theorem}(y)$ cnf(condensed_detachment, axiom)
$\text{is\_a\_theorem}(\text{implies}(x, \text{implies}(y, x)))$ cnf(mv$_1$, axiom)
$\text{is\_a\_theorem}(\text{implies}(\text{implies}(x, y), \text{implies}(\text{implies}(y, z), \text{implies}(x, z))))$ cnf(mv$_2$, axiom)
$\text{is\_a\_theorem}(\text{implies}(\text{implies}(\text{implies}(x, y), y), \text{implies}(\text{implies}(y, x), x)))$ cnf(mv$_3$, axiom)
$\text{is\_a\_theorem}(\text{implies}(\text{implies}(\text{not}(x), \text{not}(y)), \text{implies}(y, x)))$ cnf(mv$_5$, axiom)
$\neg\,\text{is\_a\_theorem}(\text{implies}(a, \text{not}(\text{not}(a))))$ cnf(prove_mv$_{29}$, negated_conjecture)

LCL112-2.p MV-29 depends on the Meredith system
An axiomatisation of the many valued sentential calculus is MV-1,MV-2,MV-3,MV-5 by Meredith. Wajsberg presented an equality axiomatisation. Show that MV-29 depends on the Wajsberg axiomatisation.
include('Axioms/LCL001-0.ax')
$\text{implies}(x, \text{not}(\text{not}(x))) \neq \text{truth}$ cnf(prove_mv$_{29}$, negated_conjecture)

LCL113-1.p MV-33 depnds on the Merideth system
An axiomatisation of the many valued sentential calculus is MV-1,MV-2,MV-3,MV-5 by Meredith. Show that 33 depends on the Meredith system.
$(\text{is\_a\_theorem}(\text{implies}(x, y)) \text{ and } \text{is\_a\_theorem}(x)) \Rightarrow \text{is\_a\_theorem}(y)$ cnf(condensed_detachment, axiom)

is_a_theorem(implies(x, implies(y, x))) cnf(mv$_1$, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z)))) cnf(mv$_2$, axiom)
is_a_theorem(implies(implies(implies(x, y), y), implies(implies(y, x), x))) cnf(mv$_3$, axiom)
is_a_theorem(implies(implies(not(x), not(y)), implies(y, x))) cnf(mv$_5$, axiom)
$\neg$ is_a_theorem(implies(implies(not(a), b), implies(not(b), a))) cnf(prove_mv$_{33}$, negated_conjecture)

LCL113-2.p MV-33 depends on the Meredith system
An axiomatisation of the many valued sentential calculus is MV-1,MV-2,MV-3,MV-5 by Meredith. Wajsberg presented an equality axiomatisation. Show that MV-33 depends on the Wajsberg axiomatisation.
include('Axioms/LCL001-0.ax')
implies(implies(not(x), y), implies(not(y), x)) $\neq$ truth cnf(prove_mv$_{33}$, negated_conjecture)

LCL114-1.p MV-36 depnds on the Merideth system
An axiomatisation of the many valued sentential calculus is MV-1,MV-2,MV-3,MV-5 by Meredith. Show that 36 depends on the Meredith system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(x, implies(y, x))) cnf(mv$_1$, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z)))) cnf(mv$_2$, axiom)
is_a_theorem(implies(implies(implies(x, y), y), implies(implies(y, x), x))) cnf(mv$_3$, axiom)
is_a_theorem(implies(implies(not(x), not(y)), implies(y, x))) cnf(mv$_5$, axiom)
$\neg$ is_a_theorem(implies(implies(a, b), implies(not(b), not(a)))) cnf(prove_mv$_{36}$, negated_conjecture)

LCL114-2.p MV-36 depends on the Meredith system
An axiomatisation of the many valued sentential calculus is MV-1,MV-2,MV-3,MV-5 by Meredith. Wajsberg presented an equality axiomatisation. Show that MV-36 depends on the Wajsberg axiomatisation.
include('Axioms/LCL001-0.ax')
implies(implies(x, y), implies(not(y), not(x))) $\neq$ truth cnf(prove_mv$_{36}$, negated_conjecture)

LCL115-1.p MV-39 depnds on the Merideth system
An axiomatisation of the many valued sentential calculus is MV-1,MV-2,MV-3,MV-5 by Meredith. Show that 39 depends on the Meredith system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(x, implies(y, x))) cnf(mv$_1$, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z)))) cnf(mv$_2$, axiom)
is_a_theorem(implies(implies(implies(x, y), y), implies(implies(y, x), x))) cnf(mv$_3$, axiom)
is_a_theorem(implies(implies(not(x), not(y)), implies(y, x))) cnf(mv$_5$, axiom)
$\neg$ is_a_theorem(implies(not(implies(a, b)), not(b))) cnf(prove_mv$_{39}$, negated_conjecture)

LCL115-2.p MV-39 depends on the Meredith system
An axiomatisation of the many valued sentential calculus is MV-1,MV-2,MV-3,MV-5 by Meredith. Wajsberg presented an equality axiomatisation. Show that MV-39 depends on the Wajsberg axiomatisation.
include('Axioms/LCL001-0.ax')
implies(not(implies(a, b)), not(b)) $\neq$ truth cnf(prove_mv$_{39}$, negated_conjecture)

LCL116-1.p MV-50 depnds on the Merideth system
An axiomatisation of the many valued sentential calculus is MV-1,MV-2,MV-3,MV-5 by Meredith. Show that 50 depends on the Meredith system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(x, implies(y, x))) cnf(mv$_1$, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z)))) cnf(mv$_2$, axiom)
is_a_theorem(implies(implies(implies(x, y), y), implies(implies(y, x), x))) cnf(mv$_3$, axiom)
is_a_theorem(implies(implies(not(x), not(y)), implies(y, x))) cnf(mv$_5$, axiom)
$\neg$ is_a_theorem(implies(not(a), implies(b, not(implies(b, a))))) cnf(prove_mv$_{50}$, negated_conjecture)

LCL116-2.p MV-50 depends on the Meredith system
An axiomatisation of the many valued sentential calculus is MV-1,MV-2,MV-3,MV-5 by Meredith. Wajsberg presented an equality axiomatisation. Show that MV-50 depends on the Wajsberg axiomatisation.
include('Axioms/LCL001-0.ax')
implies(not(a), implies(b, not(implies(b, a)))) $\neq$ truth cnf(prove_mv$_{50}$, negated_conjecture)

LCL117-1.p QYF depends on YQM
Single axioms for the R calculus are QYF, YQM, WO, all by Meredith and XGJ by Winker. Show that QYF depends on YQM.
(is_a_theorem(equivalent(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)

$\text{is\_a\_theorem}(\text{equivalent}(\text{equivalent}(x,y),\text{equivalent}(\text{equivalent}(z,y),\text{equivalent}(z,x))))$ cnf(yqm, axiom)
$\neg\,\text{is\_a\_theorem}(\text{equivalent}(\text{equivalent}(\text{equivalent}(a,b),\text{equivalent}(a,c)),\text{equivalent}(c,b)))$ cnf(prove_qyf, negated_conjecture

LCL118-1.p YQM depends on WO
Single axioms for the R calculus are QYF, YQM, WO, all by Meredith and XGJ by Winker. Show that YQM depends on WO.
$(\text{is\_a\_theorem}(\text{equivalent}(x,y)) \text{ and } \text{is\_a\_theorem}(x)) \Rightarrow \text{is\_a\_theorem}(y)$ cnf(condensed_detachment, axiom)
$\text{is\_a\_theorem}(\text{equivalent}(\text{equivalent}(x,\text{equivalent}(y,z)),\text{equivalent}(z,\text{equivalent}(y,x))))$ cnf(wo, axiom)
$\neg\,\text{is\_a\_theorem}(\text{equivalent}(\text{equivalent}(a,b),\text{equivalent}(\text{equivalent}(c,b),\text{equivalent}(c,a))))$ cnf(prove_yqm, negated_conjectu

LCL119-1.p WO depends on XGJ
Single axioms for the R calculus are QYF, YQM, WO, all by Meredith and XGJ by Winker. Show that WO depends on XGJ.
$(\text{is\_a\_theorem}(\text{equivalent}(x,y)) \text{ and } \text{is\_a\_theorem}(x)) \Rightarrow \text{is\_a\_theorem}(y)$ cnf(condensed_detachment, axiom)
$\text{is\_a\_theorem}(\text{equivalent}(x,\text{equivalent}(\text{equivalent}(y,\text{equivalent}(z,x)),\text{equivalent}(y,z))))$ cnf(xgj, axiom)
$\neg\,\text{is\_a\_theorem}(\text{equivalent}(\text{equivalent}(a,\text{equivalent}(b,c)),\text{equivalent}(c,\text{equivalent}(b,a))))$ cnf(prove_wo, negated_conjecture

LCL120-1.p XGJ depends on QYF
Single axioms for the R calculus are QYF, YQM, WO, all by Meredith and XGJ by Winker. Show that XGJ depends on QYF.
$(\text{is\_a\_theorem}(\text{equivalent}(x,y)) \text{ and } \text{is\_a\_theorem}(x)) \Rightarrow \text{is\_a\_theorem}(y)$ cnf(condensed_detachment, axiom)
$\text{is\_a\_theorem}(\text{equivalent}(\text{equivalent}(\text{equivalent}(x,y),\text{equivalent}(x,z)),\text{equivalent}(z,y)))$ cnf(qyf, axiom)
$\neg\,\text{is\_a\_theorem}(\text{equivalent}(a,\text{equivalent}(\text{equivalent}(b,\text{equivalent}(c,a)),\text{equivalent}(b,c))))$ cnf(prove_xgj, negated_conjecture

LCL121-1.p LG-1 depends on LG-2
Kalman's axiomatisation of the right group calculus is LG-1,LG-2,LG-3,LG-4,LG-5. McCune has shown that LG-2 is a single axiom. Other axiomatisations by McCune are Q-2,Q-3, Q-3,Q-4, S-2, S-3, S-4, P-4, S-6. Show that LG-1 depends on LG-2.
$(\text{is\_a\_theorem}(\text{equivalent}(x,y)) \text{ and } \text{is\_a\_theorem}(x)) \Rightarrow \text{is\_a\_theorem}(y)$ cnf(condensed_detachment, axiom)
$\text{is\_a\_theorem}(\text{equivalent}(x,\text{equivalent}(x,\text{equivalent}(\text{equivalent}(y,z),\text{equivalent}(\text{equivalent}(y,u),\text{equivalent}(z,u))))))$ cnf(lg
$\neg\,\text{is\_a\_theorem}(\text{equivalent}(a,\text{equivalent}(a,\text{equivalent}(\text{equivalent}(b,\text{equivalent}(c,c)),b))))$ cnf(prove_lg_1, negated_conjecture

LCL122-1.p LG-3 depends on LG-2
Kalman's axiomatisation of the right group calculus is LG-1,LG-2,LG-3,LG-4,LG-5. McCune has shown that LG-2 is a single axiom. Other axiomatisations by McCune are Q-2,Q-3, Q-3,Q-4, S-2, S-3, S-4, P-4, S-6. Show that LG-3 depends on LG-2.
$(\text{is\_a\_theorem}(\text{equivalent}(x,y)) \text{ and } \text{is\_a\_theorem}(x)) \Rightarrow \text{is\_a\_theorem}(y)$ cnf(condensed_detachment, axiom)
$\text{is\_a\_theorem}(\text{equivalent}(x,\text{equivalent}(x,\text{equivalent}(\text{equivalent}(y,z),\text{equivalent}(\text{equivalent}(y,u),\text{equivalent}(z,u))))))$ cnf(lg
$\neg\,\text{is\_a\_theorem}(\text{equivalent}(a,\text{equivalent}(a,\text{equivalent}(\text{equivalent}(b,\text{equivalent}(c,e)),\text{equivalent}(b,\text{equivalent}(\text{equivalent}(c,f),$ e

LCL123-1.p LG-4 depends on LG-2
Kalman's axiomatisation of the right group calculus is LG-1,LG-2,LG-3,LG-4,LG-5. McCune has shown that LG-2 is a single axiom. Other axiomatisations by McCune are Q-2,Q-3, Q-3,Q-4, S-2, S-3, S-4, P-4, S-6. Show that LG-4 depends on LG-2.
$(\text{is\_a\_theorem}(\text{equivalent}(x,y)) \text{ and } \text{is\_a\_theorem}(x)) \Rightarrow \text{is\_a\_theorem}(y)$ cnf(condensed_detachment, axiom)
$\text{is\_a\_theorem}(\text{equivalent}(x,\text{equivalent}(x,\text{equivalent}(\text{equivalent}(y,z),\text{equivalent}(\text{equivalent}(y,u),\text{equivalent}(z,u))))))$ cnf(lg
$\neg\,\text{is\_a\_theorem}(\text{equivalent}(\text{equivalent}(\text{equivalent}(a,\text{equivalent}(b,c)),\text{equivalent}(e,\text{equivalent}(b,f))),\text{equivalent}(a,\text{equivalent}(e,$

LCL124-1.p LG-5 depends on LG-2
Kalman's axiomatisation of the right group calculus is LG-1,LG-2,LG-3,LG-4,LG-5. McCune has shown that LG-2 is a single axiom. Other axiomatisations by McCune are Q-2,Q-3, Q-3,Q-4, S-2, S-3, S-4, P-4, S-6. Show that LG-5 depends on LG-2.
$(\text{is\_a\_theorem}(\text{equivalent}(x,y)) \text{ and } \text{is\_a\_theorem}(x)) \Rightarrow \text{is\_a\_theorem}(y)$ cnf(condensed_detachment, axiom)
$\text{is\_a\_theorem}(\text{equivalent}(x,\text{equivalent}(x,\text{equivalent}(\text{equivalent}(y,z),\text{equivalent}(\text{equivalent}(y,u),\text{equivalent}(z,u))))))$ cnf(lg
$\neg\,\text{is\_a\_theorem}(\text{equivalent}(\text{equivalent}(a,\text{equivalent}(b,\text{equivalent}(c,\text{equivalent}(e,f)))),\text{equivalent}(\text{equivalent}(a,\text{equivalent}(f,c)$

LCL125-1.p LG-2 depends on the 1st McCune system
Kalman's axiomatisation of the right group calculus is LG-1,LG-2,LG-3,LG-4,LG-5. McCune has shown that LG-2 is a single axiom. Other axiomatisations by McCune are Q-2,Q-3, Q-3,Q-4, S-2, S-3, S-4, P-4, S-6. Show that LG-2 depends on the first McCune system.
$(\text{is\_a\_theorem}(\text{equivalent}(x,y)) \text{ and } \text{is\_a\_theorem}(x)) \Rightarrow \text{is\_a\_theorem}(y)$ cnf(condensed_detachment, axiom)
$\text{is\_a\_theorem}(\text{equivalent}(\text{equivalent}(\text{equivalent}(x,y),\text{equivalent}(z,y)),\text{equivalent}(x,z)))$ cnf(q_2, axiom)
$\text{is\_a\_theorem}(\text{equivalent}(x,\text{equivalent}(\text{equivalent}(x,\text{equivalent}(y,z)),\text{equivalent}(z,y))))$ cnf(q_3, axiom)
$\neg\,\text{is\_a\_theorem}(\text{equivalent}(a,\text{equivalent}(a,\text{equivalent}(\text{equivalent}(b,c),\text{equivalent}(\text{equivalent}(b,e),\text{equivalent}(c,e))))))$ cnf(p

LCL126-1.p Q-2 depends on the 2nd McCune system
Kalman's axiomatisation of the right group calculus is LG-1,LG-2,LG-3,LG-4,LG-5. McCune has shown that LG-2 is a single axiom. Other axiomatisations by McCune are Q-2,Q-3, Q-3,Q-4, S-2, S-3, S-4, P-4, S-6. Show that Q-2 depends on the second McCune system.
(is_a_theorem(equivalent(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(equivalent(x, equivalent(equivalent(x, equivalent(y, z)), equivalent(z, y))))   cnf(q_3, axiom)
is_a_theorem(equivalent(equivalent(x, y), equivalent(equivalent(x, z), equivalent(y, z))))   cnf(q_4, axiom)
$\neg$ is_a_theorem(equivalent(equivalent(equivalent(a, b), equivalent(c, b)), equivalent(a, c)))   cnf(prove_q_2, negated_conjecture

LCL127-1.p LG-2 depends on S-2
Kalman's axiomatisation of the right group calculus is LG-1,LG-2,LG-3,LG-4,LG-5. McCune has shown that LG-2 is a single axiom. Other axiomatisations by McCune are Q-2,Q-3, Q-3,Q-4, S-2, S-3, S-4, P-4, S-6. Show that LG-2 depends on S-2.
(is_a_theorem(equivalent(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(equivalent(equivalent(x, equivalent(y, z)), equivalent(x, equivalent(equivalent(y, u), equivalent(z, u)))))   cnf(s
$\neg$ is_a_theorem(equivalent(a, equivalent(a, equivalent(equivalent(b, c), equivalent(equivalent(b, e), equivalent(c, e))))))   cnf(p

LCL128-1.p LG-2 depends on S-3
Kalman's axiomatisation of the right group calculus is LG-1,LG-2,LG-3,LG-4,LG-5. McCune has shown that LG-2 is a single axiom. Other axiomatisations by McCune are Q-2,Q-3, Q-3,Q-4, S-2, S-3, S-4, P-4, S-6. Show that LG-2 depends on S-3.
(is_a_theorem(equivalent(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(equivalent(x, equivalent(x, equivalent(equivalent(equivalent(y, z), equivalent(u, z)), equivalent(y, u)))))   cnf(s
$\neg$ is_a_theorem(equivalent(a, equivalent(a, equivalent(equivalent(b, c), equivalent(equivalent(b, e), equivalent(c, e))))))   cnf(p

LCL129-1.p LG-2 depends on S-4
Kalman's axiomatisation of the right group calculus is LG-1,LG-2,LG-3,LG-4,LG-5. McCune has shown that LG-2 is a single axiom. Other axiomatisations by McCune are Q-2,Q-3, Q-3,Q-4, S-2, S-3, S-4, P-4, S-6. Show that LG-2 depends on S-4.
(is_a_theorem(equivalent(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(equivalent(equivalent(x, equivalent(y, z)), equivalent(equivalent(x, equivalent(u, z)), equivalent(y, u))))   cnf(s
$\neg$ is_a_theorem(equivalent(a, equivalent(a, equivalent(equivalent(b, c), equivalent(equivalent(b, e), equivalent(c, e))))))   cnf(p

LCL130-1.p LG-2 depends on P-4
Kalman's axiomatisation of the right group calculus is LG-1,LG-2,LG-3,LG-4,LG-5. McCune has shown that LG-2 is a single axiom. Other axiomatisations by McCune are Q-2,Q-3, Q-3,Q-4, S-2, S-3, S-4, P-4, S-6. Show that LG-2 depends on P-4.
(is_a_theorem(equivalent(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(equivalent(equivalent(x, equivalent(equivalent(y, z), equivalent(equivalent(y, u), equivalent(z, u)))), x))   cnf(p
$\neg$ is_a_theorem(equivalent(a, equivalent(a, equivalent(equivalent(b, c), equivalent(equivalent(b, e), equivalent(c, e))))))   cnf(p

LCL131-1.p LG-2 depends on S-6
Kalman's axiomatisation of the right group calculus is LG-1,LG-2,LG-3,LG-4,LG-5. McCune has shown that LG-2 is a single axiom. Other axiomatisations by McCune are Q-2,Q-3, Q-3,Q-4, S-2, S-3, S-4, P-4, S-6. Show that LG-2 depends on S-6.
(is_a_theorem(equivalent(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(equivalent(equivalent(x, equivalent(equivalent(equivalent(y, z), equivalent(u, z)), equivalent(y, u))), x))   cnf(s
$\neg$ is_a_theorem(equivalent(a, equivalent(a, equivalent(equivalent(b, c), equivalent(equivalent(b, e), equivalent(c, e))))))   cnf(p

LCL132-1.p A lemma in Wajsberg algebras
include('Axioms/LCL001-0.ax')
implies(x, x) $\neq$ truth   cnf(prove_wajsberg_lemma, negated_conjecture)

LCL133-1.p A lemma in Wajsberg algebras
include('Axioms/LCL001-0.ax')
implies(x, y) = implies(y, x)   cnf(lemma_antecedent, negated_conjecture)
$x \neq y$ cnf(prove_wajsberg_lemma, negated_conjecture)

LCL134-1.p A lemma in Wajsberg algebras
include('Axioms/LCL001-0.ax')
implies(x, truth) $\neq$ truth   cnf(prove_wajsberg_lemma, negated_conjecture)

LCL135-1.p A lemma in Wajsberg algebras
An axiomatisation of the many valued sentential calculus is MV-1,MV-2,MV-3,MV-5 by Meredith. Wajsberg provided a different axiomatisation. Show that MV-1 depends on the Wajsberg system.

include('Axioms/LCL001-0.ax')
$\text{implies}(x, \text{implies}(y, x)) \neq \text{truth}$ cnf(prove_wajsberg_lemma, negated_conjecture)

LCL136-1.p A lemma in Wajsberg algebras
An axiomatisation of the many valued sentential calculus is MV-1,MV-2,MV-3,MV-5 by Meredith. Wajsberg provided a different axiomatisation. Show that a version of MV-2 depends on the Wajsberg system.
include('Axioms/LCL001-0.ax')
$\text{implies}(x, y) = \text{implies}(y, z)$ cnf(lemma_antecedent, negated_conjecture)
$\text{implies}(x, z) \neq \text{truth}$ cnf(prove_wajsberg_lemma, negated_conjecture)

LCL137-1.p A lemma in Wajsberg algebras
An axiomatisation of the many valued sentential calculus is MV-1,MV-2,MV-3,MV-5 by Meredith. Wajsberg provided a different axiomatisation. Show that MV-3 depends on the Wajsberg system.
include('Axioms/LCL001-0.ax')
$\text{implies}(\text{implies}(\text{implies}(x, y), y), \text{implies}(\text{implies}(y, z), \text{implies}(x, z))) \neq \text{truth}$ cnf(prove_wajsberg_lemma, negated_conjectu

LCL138-1.p A lemma in Wajsberg algebras
include('Axioms/LCL001-0.ax')
$\text{implies}(x, \text{implies}(y, z)) \neq \text{implies}(y, \text{implies}(x, z))$ cnf(prove_wajsberg_lemma, negated_conjecture)

LCL139-1.p A lemma in Wajsberg algebras
include('Axioms/LCL001-0.ax')
$\text{implies}(x, \text{not}(\text{truth})) \neq \text{not}(x)$ cnf(prove_wajsberg_lemma, negated_conjecture)

LCL140-1.p A lemma in Wajsberg algebras
include('Axioms/LCL001-0.ax')
$\text{not}(\text{not}(x)) \neq x$ cnf(prove_wajsberg_lemma, negated_conjecture)

LCL141-1.p A lemma in Wajsberg algebras
An axiomatisation of the many valued sentential calculus is MV-1,MV-2,MV-3,MV-5 by Meredith. Wajsberg provided a different axiomatisation. Show that MV-5 depends on the Wajsberg system.
include('Axioms/LCL001-0.ax')
$\text{implies}(\text{not}(x), \text{not}(y)) \neq \text{implies}(y, x)$ cnf(prove_wajsberg_lemma, negated_conjecture)

LCL142-1.p A theorem in the lattice structure of Wajsberg algebras
include('Axioms/LCL001-0.ax')
include('Axioms/LCL001-1.ax')
$\text{ordered}(x, y)$ cnf(antecedent, negated_conjecture)
$\neg\,\text{ordered}(\text{implies}(x, z), \text{implies}(y, z))$ cnf(prove_wajsberg_theorem, negated_conjecture)

LCL143-1.p A theorem in the lattice structure of Wajsberg algebras
include('Axioms/LCL001-0.ax')
include('Axioms/LCL001-1.ax')
$\text{ordered}(x, y)$ cnf(antecedent, negated_conjecture)
$\neg\,\text{ordered}(\text{implies}(z, x), \text{implies}(z, y))$ cnf(prove_wajsberg_theorem, negated_conjecture)

LCL144-1.p A theorem in the lattice structure of Wajsberg algebras
include('Axioms/LCL001-0.ax')
include('Axioms/LCL001-1.ax')
$\text{ordered}(x, \text{implies}(y, z))$ or $\text{ordered}(y, \text{implies}(x, z))$ cnf(antecedent, negated_conjecture)
$\text{ordered}(x, \text{implies}(y, z)) \Rightarrow \neg\,\text{ordered}(y, \text{implies}(x, z))$ cnf(prove_wajsberg_theorem, negated_conjecture)

LCL145-1.p A theorem in the lattice structure of Wajsberg algebras
include('Axioms/LCL001-0.ax')
include('Axioms/LCL001-1.ax')
$\text{not}(\text{big\_V}(x, y)) \neq \text{big\_hat}(\text{not}(x), \text{not}(y))$ cnf(prove_wajsberg_theorem, negated_conjecture)

LCL146-1.p A theorem in the lattice structure of Wajsberg algebras
include('Axioms/LCL001-0.ax')
include('Axioms/LCL001-1.ax')
$\text{not}(\text{big\_hat}(x, y)) \neq \text{big\_V}(\text{not}(x), \text{not}(y))$ cnf(prove_wajsberg_theorem, negated_conjecture)

LCL147-1.p A theorem in the lattice structure of Wajsberg algebras
include('Axioms/LCL001-0.ax')
include('Axioms/LCL001-1.ax')
$\text{implies}(\text{big\_V}(x, y), z) \neq \text{big\_hat}(\text{implies}(x, z), \text{implies}(y, z))$ cnf(prove_wajsberg_theorem, negated_conjecture)

LCL148-1.p A theorem in the lattice structure of Wajsberg algebras
include('Axioms/LCL001-0.ax')
include('Axioms/LCL001-1.ax')
$\text{implies}(x, \text{big\_hat}(y, z)) \neq \text{big\_hat}(\text{implies}(x, y), \text{implies}(x, z))$ cnf(prove_wajsberg_theorem, negated_conjecture)

LCL149-1.p A theorem in the lattice structure of Wajsberg algebras
include('Axioms/LCL001-0.ax')
include('Axioms/LCL001-1.ax')
$\text{implies}(x, \text{big\_V}(y, z)) \neq \text{big\_V}(\text{implies}(x, y), \text{implies}(x, z))$ cnf(prove_wajsberg_theorem, negated_conjecture)

LCL150-1.p A theorem in the lattice structure of Wajsberg algebras
include('Axioms/LCL001-0.ax')
include('Axioms/LCL001-1.ax')
$\text{implies}(\text{big\_hat}(x, y), z) \neq \text{big\_V}(\text{implies}(x, z), \text{implies}(y, z))$ cnf(prove_wajsberg_theorem, negated_conjecture)

LCL151-1.p A theorem in the lattice structure of Wajsberg algebras
include('Axioms/LCL001-0.ax')
include('Axioms/LCL001-1.ax')
$\text{big\_V}(\text{big\_hat}(x, y), z) \neq \text{big\_hat}(\text{big\_V}(x, z), \text{big\_V}(y, z))$ cnf(prove_wajsberg_theorem, negated_conjecture)

LCL152-1.p A theorem in the lattice structure of Wajsberg algebras
include('Axioms/LCL001-0.ax')
include('Axioms/LCL001-1.ax')
$\text{implies}(\text{big\_hat}(x, y), z) \neq \text{implies}(\text{implies}(x, y), \text{implies}(x, z))$ cnf(prove_wajsberg_theorem, negated_conjecture)

LCL153-1.p The 1st alternative Wajsberg algebra axiom
include('Axioms/LCL001-0.ax')
include('Axioms/LCL001-2.ax')
include('Axioms/LCL002-1.ax')
$\text{not}(x) \neq \text{xor}(x, \text{truth})$ cnf(prove_alternative_wajsberg_axiom, negated_conjecture)

LCL154-1.p The 2nd alternative Wajsberg algebra axiom
include('Axioms/LCL001-0.ax')
include('Axioms/LCL001-2.ax')
include('Axioms/LCL002-1.ax')
$\text{xor}(x, \text{falsehood}) \neq x$ cnf(prove_alternative_wajsberg_axiom, negated_conjecture)

LCL155-1.p The 3rd alternative Wajsberg algebra axiom
include('Axioms/LCL001-0.ax')
include('Axioms/LCL001-2.ax')
include('Axioms/LCL002-1.ax')
$\text{xor}(x, x) \neq \text{falsehood}$ cnf(prove_alternative_wajsberg_axiom, negated_conjecture)

LCL156-1.p The 4th alternative Wajsberg algebra axiom
include('Axioms/LCL001-0.ax')
include('Axioms/LCL001-2.ax')
include('Axioms/LCL002-1.ax')
$\text{and\_star}(x, \text{truth}) \neq x$ cnf(prove_alternative_wajsberg_axiom, negated_conjecture)

LCL157-1.p The 5th alternative Wajsberg algebra axiom
include('Axioms/LCL001-0.ax')
include('Axioms/LCL001-2.ax')
include('Axioms/LCL002-1.ax')
$\text{and\_star}(x, \text{falsehood}) \neq \text{falsehood}$ cnf(prove_alternative_wajsberg_axiom, negated_conjecture)

LCL158-1.p The 6th alternative Wajsberg algebra axiom
include('Axioms/LCL001-0.ax')
include('Axioms/LCL001-2.ax')
include('Axioms/LCL002-1.ax')
$\text{and\_star}(\text{xor}(\text{truth}, x), x) \neq \text{falsehood}$ cnf(prove_alternative_wajsberg_axiom, negated_conjecture)

LCL159-1.p The 7th alternative Wajsberg algebra axiom
include('Axioms/LCL001-0.ax')
include('Axioms/LCL001-2.ax')
include('Axioms/LCL002-1.ax')
$\text{xor}(x, \text{xor}(\text{truth}, y)) \neq \text{xor}(\text{xor}(x, \text{truth}), y)$ cnf(prove_alternative_wajsberg_axiom, negated_conjecture)

LCL160-1.p The 8th alternative Wajsberg algebra axiom
include('Axioms/LCL001-0.ax')
include('Axioms/LCL001-2.ax')
include('Axioms/LCL002-1.ax')
and_star(xor(and_star(xor(truth, x), y), truth), y) $\neq$ and_star(xor(and_star(xor(truth, y), x), truth), x)   cnf(prove_alternativ

LCL161-1.p The 1st Wajsberg algebra axiom, from the alternative axioms
include('Axioms/LCL002-0.ax')
xor(x, y) = xor(y, x)   cnf(xor_commutativity, axiom)
and_star(and_star(x, y), z) = and_star(x, and_star(y, z))   cnf(and_star_associativity, axiom)
and_star(x, y) = and_star(y, x)   cnf(and_star_commutativity, axiom)
not(truth) = falsehood   cnf(false_definition, axiom)
implies(x, y) = xor(truth, and_star(x, xor(truth, y)))   cnf(implies_definition, axiom)
implies(truth, x) $\neq$ x   cnf(prove_wajsberg_axiom, negated_conjecture)

LCL162-1.p The 2nd Wajsberg algebra axiom, from the alternative axioms
include('Axioms/LCL002-0.ax')
xor(x, y) = xor(y, x)   cnf(xor_commutativity, axiom)
and_star(and_star(x, y), z) = and_star(x, and_star(y, z))   cnf(and_star_associativity, axiom)
and_star(x, y) = and_star(y, x)   cnf(and_star_commutativity, axiom)
not(truth) = falsehood   cnf(false_definition, axiom)
implies(x, y) = xor(truth, and_star(x, xor(truth, y)))   cnf(implies_definition, axiom)
implies(implies(x, y), implies(implies(y, z), implies(x, z))) $\neq$ truth   cnf(prove_wajsberg_axiom, negated_conjecture)

LCL163-1.p The 3rd Wajsberg algebra axiom, from the alternative axioms
include('Axioms/LCL002-0.ax')
xor(x, y) = xor(y, x)   cnf(xor_commutativity, axiom)
and_star(and_star(x, y), z) = and_star(x, and_star(y, z))   cnf(and_star_associativity, axiom)
and_star(x, y) = and_star(y, x)   cnf(and_star_commutativity, axiom)
not(truth) = falsehood   cnf(false_definition, axiom)
implies(x, y) = xor(truth, and_star(x, xor(truth, y)))   cnf(implies_definition, axiom)
implies(implies(x, y), y) $\neq$ implies(implies(y, x), x)   cnf(prove_wajsberg_axiom, negated_conjecture)

LCL164-1.p The 4th Wajsberg algebra axiom, from the alternative axioms
include('Axioms/LCL002-0.ax')
xor(x, y) = xor(y, x)   cnf(xor_commutativity, axiom)
and_star(and_star(x, y), z) = and_star(x, and_star(y, z))   cnf(and_star_associativity, axiom)
and_star(x, y) = and_star(y, x)   cnf(and_star_commutativity, axiom)
not(truth) = falsehood   cnf(false_definition, axiom)
implies(x, y) = xor(truth, and_star(x, xor(truth, y)))   cnf(implies_definition, axiom)
implies(implies(not(x), not(y)), implies(y, x)) $\neq$ truth   cnf(prove_wajsberg_axiom, negated_conjecture)

LCL165-1.p A theorem in Wajsberg algebras
include('Axioms/LCL001-0.ax')
include('Axioms/LCL001-2.ax')
not(or(and(x, or(x, x)), and(x, x))) $\neq$ and(not(x), or(or(not(x), not(x)), and(not(x), not(x))))   cnf(prove_wajsberg_theorem

LCL166-1.p UM depends on XHN
Show that the single Meredith axiom UM can be derived from the single Winker axiom XHN.
(is_a_theorem(equivalent(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(equivalent(x, equivalent(equivalent(y, z), equivalent(equivalent(z, x), y))))   cnf(xhn, axiom)
$\neg$ is_a_theorem(equivalent(equivalent(equivalent(a, b), c), equivalent(b, equivalent(c, a))))   cnf(prove_um, negated_conjecture

LCL167-1.p YRO depends on XHK
Show that the single Meredith axiom YRO can be derived from the single Winker axiom XHK.
(is_a_theorem(equivalent(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(equivalent(x, equivalent(equivalent(y, z), equivalent(equivalent(x, z), y))))   cnf(xhk, axiom)
$\neg$ is_a_theorem(equivalent(equivalent(a, b), equivalent(c, equivalent(equivalent(c, b), a))))   cnf(prove_yro, negated_conjecture

LCL168-1.p XEH is not a single axiom for the R-calculus
To show that XEH is not a single axiom, attempt to derive from it any one of YQM, WO, XGJ or QYF, which are known single axioms.
(is_a_theorem(equivalent(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(equivalent(x, equivalent(equivalent(y, equivalent(equivalent(y, z), x)), z)))   cnf(xeh, axiom)

$\neg$ is_a_theorem(equivalent(equivalent(equivalent(a, b), equivalent(a, c)), equivalent(c, b))) cnf(try_prove_qyf, negated_conjec
$\neg$ is_a_theorem(equivalent(equivalent(a, b), equivalent(equivalent(c, b), equivalent(c, a)))) cnf(try_prove_yqm, negated_conje
$\neg$ is_a_theorem(equivalent(equivalent(a, equivalent(b, c)), equivalent(c, equivalent(b, a)))) cnf(try_prove_wo, negated_conjec
$\neg$ is_a_theorem(equivalent(a, equivalent(equivalent(b, equivalent(c, a)), equivalent(b, c)))) cnf(try_prove_xgj, negated_conjec

LCL169-1.p Principia Mathematica 2.01
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(or(not(p), not(p))), not(p))) cnf(prove_this, negated_conjecture)

LCL169-3.p Principia Mathematica 2.01
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(implies(p, not(p)), not(p))) cnf(prove_this, negated_conjecture)

LCL170-1.p Principia Mathematica 2.02
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(q), or(not(p), q))) cnf(prove_this, negated_conjecture)

LCL170-3.p Principia Mathematica 2.02
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(q, implies(p, q))) cnf(prove_this, negated_conjecture)

LCL171-1.p Principia Mathematica 2.03
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(or(not(p), not(q))), or(not(q), not(p)))) cnf(prove_this, negated_conjecture)

LCL171-3.p Principia Mathematica 2.03
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(implies(p, not(q)), implies(q, not(p)))) cnf(prove_this, negated_conjecture)

LCL172-1.p Principia Mathematica 2.04
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(or(not(p), or(not(q), r))), or(not(q), or(not(p), r)))) cnf(prove_this, negated_conjecture)

LCL172-3.p Principia Mathematica 2.04
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(implies(p, implies(q, r)), implies(q, implies(p, r)))) cnf(prove_this, negated_conjecture)

LCL173-1.p Principia Mathematica 2.05
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(or(not(q), r)), or(not(or(not(p), q)), or(not(p), r)))) cnf(prove_this, negated_conjecture)

LCL173-3.p Principia Mathematica 2.05
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(implies(q, r), implies(implies(p, q), implies(p, r)))) cnf(prove_this, negated_conjecture)

LCL174-1.p Principia Mathematica 2.06
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(or(not(p), q)), or(not(or(not(q), r)), or(not(p), r)))) cnf(prove_this, negated_conjecture)

LCL174-3.p Principia Mathematica 2.06
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(implies(p, q), implies(implies(q, r), implies(p, r)))) cnf(prove_this, negated_conjecture)

LCL175-1.p Principia Mathematica 2.07
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(p), or(p, p))) cnf(prove_this, negated_conjecture)

LCL175-3.p Principia Mathematica 2.07
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(p, or(p, p))) cnf(prove_this, negated_conjecture)

LCL176-1.p Principia Mathematica 2.1 and 2.08
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(p), p)) cnf(prove_this, negated_conjecture)

LCL176-3.p Principia Mathematica 2.1 and 2.08
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(p, p)) cnf(prove_this, negated_conjecture)

LCL177-1.p Principia Mathematica 2.11

include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(p, not(p))) cnf(prove_this, negated_conjecture)

LCL178-1.p Principia Mathematica 2.12
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(p), not(not(p)))) cnf(prove_this, negated_conjecture)

LCL178-3.p Principia Mathematica 2.12
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(p, not(not(p)))) cnf(prove_this, negated_conjecture)

LCL179-1.p Principia Mathematica 2.13
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(p, not(not(not(p))))) cnf(prove_this, negated_conjecture)

LCL180-1.p Principia Mathematica 2.14
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(not(not(p))), p)) cnf(prove_this, negated_conjecture)

LCL180-3.p Principia Mathematica 2.14
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(not(not(p)), p)) cnf(prove_this, negated_conjecture)

LCL181+1.p Principia Mathematica 2.15
Judged by [SRM73] to be the 'hardest' of the first 52 theorems of [WR27].
$(\neg p \Rightarrow q) \iff (\neg q \Rightarrow p)$ fof(pel_4, conjecture)

LCL181-1.p Principia Mathematica 2.15
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(or(not(not(p)), q)), or(not(not(q)), p))) cnf(prove_this, negated_conjecture)

LCL181-2.p Principia Mathematica 2.15
Judged by [SRM73] to be the 'hardest' of the first 52 theorems of [WR27].
p or q cnf(clause_1, negated_conjecture)
$q \Rightarrow \neg p$ cnf(clause_2, negated_conjecture)
$\neg q$ cnf(clause_3, negated_conjecture)
$\neg p$ cnf(clause_4, negated_conjecture)

LCL181-3.p Principia Mathematica 2.15
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(implies(not(p), q), implies(not(q), p))) cnf(prove_this, negated_conjecture)

LCL181^4.p Principia Mathematica 2.15
include('Axioms/LCL010^0.ax')
p: \$i → \$o thf(p_type, type)
q: \$i → \$o thf(q_type, type)
ivalid@(iequiv@(iimplies@(inot@(iatom@p))@(iatom@q))@(iimplies@(inot@(iatom@q))@(iatom@p))) thf(pel_4, conjecture

LCL182-1.p Principia Mathematica 2.16
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(or(not(p), q)), or(not(not(q)), not(p)))) cnf(prove_this, negated_conjecture)

LCL182-3.p Principia Mathematica 2.16
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(implies(p, q), implies(not(q), not(p)))) cnf(prove_this, negated_conjecture)

LCL183-1.p Principia Mathematica 2.17
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(or(not(not(q)), not(p))), or(not(p), q))) cnf(prove_this, negated_conjecture)

LCL183-3.p Principia Mathematica 2.17
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(implies(not(q), not(p)), implies(p, q))) cnf(prove_this, negated_conjecture)

LCL184-1.p Principia Mathematica 2.18
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(or(not(not(p)), p)), p)) cnf(prove_this, negated_conjecture)

LCL185-1.p Principia Mathematica 2.2

include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(p), or(p, q))) cnf(prove_this, negated_conjecture)

LCL185-3.p Principia Mathematica 2.2
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(p, or(p, q))) cnf(prove_this, negated_conjecture)

LCL186-1.p Principia Mathematica 2.21
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(not(p)), or(not(p), q))) cnf(prove_this, negated_conjecture)

LCL186-3.p Principia Mathematica 2.21
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(not(p), implies(p, q))) cnf(prove_this, negated_conjecture)

LCL187-1.p Principia Mathematica 2.24
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(p), or(not(not(p)), q))) cnf(prove_this, negated_conjecture)

LCL187-3.p Principia Mathematica 2.24
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(p, implies(not(p), q))) cnf(prove_this, negated_conjecture)

LCL188-1.p Principia Mathematica 2.25
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(p, or(not(or(p, q)), q))) cnf(prove_this, negated_conjecture)

LCL188-3.p Principia Mathematica 2.25
include('Axioms/LCL004-0.ax')
$\neg$ theorem(or(p, implies(or(p, q), q))) cnf(prove_this, negated_conjecture)

LCL189-1.p Principia Mathematica 2.26 and 2.27
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(p), or(not(or(not(p), q)), q))) cnf(prove_this, negated_conjecture)

LCL189-3.p Principia Mathematica 2.26
include('Axioms/LCL004-0.ax')
$\neg$ theorem(or(not(p), implies(implies(p, q), q))) cnf(prove_this, negated_conjecture)

LCL190-1.p Principia Mathematica 2.3
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(or(p, or(q, r))), or(p, or(r, q)))) cnf(prove_this, negated_conjecture)

LCL190-3.p Principia Mathematica 2.3
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(or(p, or(q, r)), or(p, or(r, q)))) cnf(prove_this, negated_conjecture)

LCL191-1.p Principia Mathematica 2.31
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(or(p, or(q, r))), or(or(p, q), r))) cnf(prove_this, negated_conjecture)

LCL191-3.p Principia Mathematica 2.31
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(or(p, or(q, r)), or(or(p, q), r))) cnf(prove_this, negated_conjecture)

LCL192-1.p Principia Mathematica 2.32 and 2.33
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(or(or(p, q), r)), or(p, or(q, r)))) cnf(prove_this, negated_conjecture)

LCL192-3.p Principia Mathematica 2.32
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(or(or(p, q), r), or(p, or(q, r)))) cnf(prove_this, negated_conjecture)

LCL193-1.p Principia Mathematica 2.36
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(or(not(q), r)), or(not(or(p, q)), or(r, p)))) cnf(prove_this, negated_conjecture)

LCL193-3.p Principia Mathematica 2.36
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(implies(q, r), implies(or(p, q), or(r, p)))) cnf(prove_this, negated_conjecture)

LCL194-1.p Principia Mathematica 2.37
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(or(not(q), r)), or(not(or(q, p)), or(p, r)))) cnf(prove_this, negated_conjecture)

LCL194-3.p Principia Mathematica 2.37
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(implies(q, r), implies(or(q, p), or(p, r)))) cnf(prove_this, negated_conjecture)

LCL195-1.p Principia Mathematica 2.38
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(or(not(q), r)), or(not(or(q, p)), or(r, p)))) cnf(prove_this, negated_conjecture)

LCL195-3.p Principia Mathematica 2.38
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(implies(q, r), implies(or(q, p), or(r, p)))) cnf(prove_this, negated_conjecture)

LCL196-1.p Principia Mathematica 2.4
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(or(p, or(p, q))), or(p, q))) cnf(prove_this, negated_conjecture)

LCL196-3.p Principia Mathematica 2.4
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(or(p, or(p, q)), or(p, q))) cnf(prove_this, negated_conjecture)

LCL197-1.p Principia Mathematica 2.41
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(or(q, or(p, q))), or(p, q))) cnf(prove_this, negated_conjecture)

LCL197-3.p Principia Mathematica 2.41
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(or(q, or(p, q)), or(p, q))) cnf(prove_this, negated_conjecture)

LCL198-1.p Principia Mathematica 2.42 and 2.43
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(or(not(p), or(not(p), q))), or(not(p), q))) cnf(prove_this, negated_conjecture)

LCL198-3.p Principia Mathematica 2.42
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(or(not(p), implies(p, q)), implies(p, q))) cnf(prove_this, negated_conjecture)

LCL199-1.p Principia Mathematica 2.45
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(not(or(p, q))), not(p))) cnf(prove_this, negated_conjecture)

LCL199-3.p Principia Mathematica 2.45
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(not(or(p, q)), not(p))) cnf(prove_this, negated_conjecture)

LCL200-1.p Principia Mathematica 2.46
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(not(or(p, q))), not(q))) cnf(prove_this, negated_conjecture)

LCL201-1.p Principia Mathematica 2.47
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(not(or(p, q))), or(not(p), q))) cnf(prove_this, negated_conjecture)

LCL201-3.p Principia Mathematica 2.47
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(not(or(p, q)), or(not(p), q))) cnf(prove_this, negated_conjecture)

LCL202-1.p Principia Mathematica 2.48
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(not(or(p, q))), or(p, not(q)))) cnf(prove_this, negated_conjecture)

LCL202-3.p Principia Mathematica 2.48
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(not(or(p, q)), or(p, not(q)))) cnf(prove_this, negated_conjecture)

LCL203-1.p Principia Mathematica 2.49
include('Axioms/LCL003-0.ax')

$\neg$ theorem(or(not(not(or(p,q))), or(not(p), not(q))))   cnf(prove_this, negated_conjecture)

LCL203-3.p Principia Mathematica 2.49
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(not(or(p,q)), or(not(p), not(q))))   cnf(prove_this, negated_conjecture)

LCL204-1.p Principia Mathematica 2.5
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(not(or(not(p), q))), or(not(not(p)), q)))   cnf(prove_this, negated_conjecture)

LCL205-1.p Principia Mathematica 2.51
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(not(or(not(p), q))), or(not(p), not(q))))   cnf(prove_this, negated_conjecture)

LCL205-3.p Principia Mathematica 2.51
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(not(implies(p,q)), implies(p, not(q))))   cnf(prove_this, negated_conjecture)

LCL206-1.p Principia Mathematica 2.52
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(not(or(not(p), q))), or(not(not(p)), not(q))))   cnf(prove_this, negated_conjecture)

LCL206-3.p Principia Mathematica 2.52
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(not(implies(p,q)), implies(not(p), not(q))))   cnf(prove_this, negated_conjecture)

LCL207-1.p Principia Mathematica 2.521
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(not(or(not(p), q))), or(not(q), p)))   cnf(prove_this, negated_conjecture)

LCL207-3.p Principia Mathematica 2.521
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(not(implies(p,q)), implies(q,p)))   cnf(prove_this, negated_conjecture)

LCL208-1.p Principia Mathematica 2.53
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(or(p,q)), or(not(not(p)), q)))   cnf(prove_this, negated_conjecture)

LCL208-3.p Principia Mathematica 2.53
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(or(p,q), implies(not(p), q)))   cnf(prove_this, negated_conjecture)

LCL209-1.p Principia Mathematica 2.54
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(or(not(not(p)), q)), or(p,q)))   cnf(prove_this, negated_conjecture)

LCL209-3.p Principia Mathematica 2.54
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(implies(not(p), q), or(p,q)))   cnf(prove_this, negated_conjecture)

LCL210-1.p Principia Mathematica 2.55
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(not(p)), or(not(or(p,q)), q)))   cnf(prove_this, negated_conjecture)

LCL210-3.p Principia Mathematica 2.55
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(not(p), implies(or(p,q), q)))   cnf(prove_this, negated_conjecture)

LCL211-1.p Principia Mathematica 2.56
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(not(q)), or(not(or(p,q)), p)))   cnf(prove_this, negated_conjecture)

LCL211-3.p Principia Mathematica 2.56
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(not(q), implies(or(p,q), p)))   cnf(prove_this, negated_conjecture)

LCL212-1.p Principia Mathematica 2.6
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(or(not(not(p)), q)), or(not(or(not(p), q)), q)))   cnf(prove_this, negated_conjecture)

LCL212-3.p Principia Mathematica 2.6

include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(not(p), implies(q, implies(implies(p, q), q)))) cnf(prove_this, negated_conjecture)

LCL213-1.p Principia Mathematica 2.61
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(or(not(p), q)), or(not(or(not(not(p)), q)), q))) cnf(prove_this, negated_conjecture)

LCL213-3.p Principia Mathematica 2.61
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(implies(p, q), implies(implies(not(p), q), q))) cnf(prove_this, negated_conjecture)

LCL214-1.p Principia Mathematica 2.61
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(or(p, q)), or(not(or(not(p), q)), q))) cnf(prove_this, negated_conjecture)

LCL214-3.p Principia Mathematica 2.61
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(or(p, q), implies(implies(p, q), q))) cnf(prove_this, negated_conjecture)

LCL215-1.p Principia Mathematica 2.62 and 2.63
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(or(not(p), q)), or(not(or(p, q)), q))) cnf(prove_this, negated_conjecture)

LCL215-3.p Principia Mathematica 2.621
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(implies(p, q), implies(or(p, q), q))) cnf(prove_this, negated_conjecture)

LCL216-1.p Principia Mathematica 2.64
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(or(p, q)), or(not(or(p, not(q))), p))) cnf(prove_this, negated_conjecture)

LCL216-3.p Principia Mathematica 2.64
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(or(p, q), implies(or(p, not(q)), p))) cnf(prove_this, negated_conjecture)

LCL217-1.p Principia Mathematica 2.65
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(or(not(p), q)), or(not(or(not(p), not(q))), not(p)))) cnf(prove_this, negated_conjecture)

LCL217-3.p Principia Mathematica 2.65
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(implies(p, q), implies(implies(p, not(q)), not(p)))) cnf(prove_this, negated_conjecture)

LCL218-1.p Principia Mathematica 2.67
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(or(not(or(p, q)), q)), or(not(p), q))) cnf(prove_this, negated_conjecture)

LCL218-3.p Principia Mathematica 2.67
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(implies(or(p, q), q), implies(p, q))) cnf(prove_this, negated_conjecture)

LCL219-1.p Principia Mathematica 2.68
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(or(not(or(not(p), q)), q)), or(p, q))) cnf(prove_this, negated_conjecture)

LCL219-3.p Principia Mathematica 2.68
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(implies(implies(p, q), q), or(p, q))) cnf(prove_this, negated_conjecture)

LCL220-1.p Principia Mathematica 2.69
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(or(not(or(not(p), q)), q)), or(not(or(not(q), p)), p))) cnf(prove_this, negated_conjecture)

LCL220-3.p Principia Mathematica 2.69
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(implies(implies(p, q), q), implies(implies(q, p), p))) cnf(prove_this, negated_conjecture)

LCL221-1.p Principia Mathematica 2.73
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(or(not(p), q)), or(not(or(or(p, q), r)), or(q, r)))) cnf(prove_this, negated_conjecture)

LCL221-3.p Principia Mathematica 2.73
include('Axioms/LCL004-0.ax')
$\neg\,\mathrm{theorem}(\mathrm{implies}(\mathrm{implies}(p,q),\mathrm{implies}(\mathrm{or}(\mathrm{or}(p,q),r),\mathrm{or}(q,r))))$ cnf(prove_this, negated_conjecture)

LCL222-1.p Principia Mathematica 2.74
include('Axioms/LCL003-0.ax')
$\neg\,\mathrm{theorem}(\mathrm{or}(\mathrm{not}(\mathrm{or}(\mathrm{not}(q),p)),\mathrm{or}(\mathrm{not}(\mathrm{or}(\mathrm{or}(p,q),r)),\mathrm{or}(p,r))))$ cnf(prove_this, negated_conjecture)

LCL222-3.p Principia Mathematica 2.74
include('Axioms/LCL004-0.ax')
$\neg\,\mathrm{theorem}(\mathrm{implies}(\mathrm{implies}(q,p),\mathrm{implies}(\mathrm{or}(\mathrm{or}(p,q),r),\mathrm{or}(p,r))))$ cnf(prove_this, negated_conjecture)

LCL223-1.p Principia Mathematica 2.75
include('Axioms/LCL003-0.ax')
$\neg\,\mathrm{theorem}(\mathrm{or}(\mathrm{not}(\mathrm{or}(p,q)),\mathrm{or}(\mathrm{not}(\mathrm{or}(p,\mathrm{or}(\mathrm{not}(q),r))),\mathrm{or}(p,r))))$ cnf(prove_this, negated_conjecture)

LCL223-3.p Principia Mathematica 2.75
include('Axioms/LCL004-0.ax')
$\neg\,\mathrm{theorem}(\mathrm{implies}(\mathrm{or}(p,q),\mathrm{implies}(\mathrm{or}(p,\mathrm{implies}(q,r)),\mathrm{or}(p,r))))$ cnf(prove_this, negated_conjecture)

LCL224-1.p Principia Mathematica 2.76
include('Axioms/LCL003-0.ax')
$\neg\,\mathrm{theorem}(\mathrm{or}(\mathrm{not}(\mathrm{or}(p,\mathrm{or}(\mathrm{not}(q),r))),\mathrm{or}(\mathrm{not}(\mathrm{or}(p,q)),\mathrm{or}(p,r))))$ cnf(prove_this, negated_conjecture)

LCL224-3.p Principia Mathematica 2.76
include('Axioms/LCL004-0.ax')
$\neg\,\mathrm{theorem}(\mathrm{implies}(\mathrm{or}(p,\mathrm{implies}(q,r)),\mathrm{implies}(\mathrm{or}(p,q),\mathrm{or}(p,r))))$ cnf(prove_this, negated_conjecture)

LCL225-1.p Principia Mathematica 2.77
include('Axioms/LCL003-0.ax')
$\neg\,\mathrm{theorem}(\mathrm{or}(\mathrm{not}(\mathrm{or}(\mathrm{not}(p),\mathrm{or}(\mathrm{not}(q),r))),\mathrm{or}(\mathrm{not}(\mathrm{or}(\mathrm{not}(p),q)),\mathrm{or}(\mathrm{not}(p),r))))$ cnf(prove_this, negated_conjecture)

LCL225-3.p Principia Mathematica 2.77
include('Axioms/LCL004-0.ax')
$\neg\,\mathrm{theorem}(\mathrm{implies}(\mathrm{implies}(p,\mathrm{implies}(q,r)),\mathrm{implies}(\mathrm{implies}(p,q),\mathrm{implies}(p,r))))$ cnf(prove_this, negated_conjecture)

LCL226-1.p Principia Mathematica 2.8
include('Axioms/LCL003-0.ax')
$\neg\,\mathrm{theorem}(\mathrm{or}(\mathrm{not}(\mathrm{or}(q,r)),\mathrm{or}(\mathrm{not}(\mathrm{or}(\mathrm{not}(r),s)),\mathrm{or}(q,s))))$ cnf(prove_this, negated_conjecture)

LCL226-3.p Principia Mathematica 2.8
include('Axioms/LCL004-0.ax')
$\neg\,\mathrm{theorem}(\mathrm{implies}(\mathrm{or}(q,r),\mathrm{implies}(\mathrm{or}(\mathrm{not}(r),s),\mathrm{or}(q,s))))$ cnf(prove_this, negated_conjecture)

LCL227-1.p Principia Mathematica 2.81
include('Axioms/LCL003-0.ax')
$\neg\,\mathrm{theorem}(\mathrm{or}(\mathrm{not}(\mathrm{or}(\mathrm{not}(q),\mathrm{or}(\mathrm{not}(r),s))),\mathrm{or}(\mathrm{not}(\mathrm{or}(p,q)),\mathrm{or}(\mathrm{not}(\mathrm{or}(p,r)),\mathrm{or}(p,s)))))$ cnf(prove_this, negated_conjecture

LCL227-3.p Principia Mathematica 2.81
include('Axioms/LCL004-0.ax')
$\neg\,\mathrm{theorem}(\mathrm{implies}(\mathrm{implies}(q,\mathrm{implies}(r,s)),\mathrm{implies}(\mathrm{or}(p,q),\mathrm{implies}(\mathrm{or}(p,r),\mathrm{or}(p,s)))))$ cnf(prove_this, negated_conjecture)

LCL228-1.p Principia Mathematica 2.82
include('Axioms/LCL003-0.ax')
$\neg\,\mathrm{theorem}(\mathrm{or}(\mathrm{not}(\mathrm{or}(\mathrm{or}(p,q),r)),\mathrm{or}(\mathrm{not}(\mathrm{or}(\mathrm{or}(p,\mathrm{not}(r)),s)),\mathrm{or}(\mathrm{or}(p,q),s))))$ cnf(prove_this, negated_conjecture)

LCL228-3.p Principia Mathematica 2.82
include('Axioms/LCL004-0.ax')
$\neg\,\mathrm{theorem}(\mathrm{implies}(\mathrm{or}(\mathrm{or}(p,q),r),\mathrm{implies}(\mathrm{or}(\mathrm{or}(p,\mathrm{not}(r)),s),\mathrm{or}(\mathrm{or}(p,q),s))))$ cnf(prove_this, negated_conjecture)

LCL229-1.p Principia Mathematica 2.83
include('Axioms/LCL003-0.ax')
$\neg\,\mathrm{theorem}(\mathrm{or}(\mathrm{not}(\mathrm{or}(\mathrm{not}(p),\mathrm{or}(\mathrm{not}(q),r))),\mathrm{or}(\mathrm{not}(\mathrm{or}(\mathrm{not}(p),\mathrm{or}(\mathrm{not}(r),s))),\mathrm{or}(\mathrm{not}(p),\mathrm{or}(\mathrm{not}(q),s)))))$ cnf(prove_this, neg

LCL229-3.p Principia Mathematica 2.83
include('Axioms/LCL004-0.ax')
$\neg\,\mathrm{theorem}(\mathrm{implies}(\mathrm{implies}(p,\mathrm{implies}(q,r)),\mathrm{implies}(\mathrm{implies}(p,\mathrm{implies}(r,s)),\mathrm{implies}(p,\mathrm{implies}(q,s)))))$ cnf(prove_this, nega

LCL230+1.p Principia Mathematica 2.85
Judged by [SRM73] to be the 'hardest' of the first 67 theorems of [WR27].

$((p \text{ or } q) \Rightarrow (p \text{ or } r)) \Rightarrow (p \text{ or } (q \Rightarrow r))$ fof(pel_5, conjecture)

LCL230-1.p Principia Mathematica 2.85
include('Axioms/LCL003-0.ax')
¬ theorem(or(not(or(not(or(p, q)), or(p, r))), or(p, or(not(q), r)))) cnf(prove_this, negated_conjecture)

LCL230-2.p Principia Mathematica 2.85
Judged by [SRM73] to be the 'hardest' of the first 67 theorems of [WR27].
$q \Rightarrow (p \text{ or } r)$ cnf(clause_1, negated_conjecture)
$\neg p$ cnf(clause_2, negated_conjecture)
q cnf(clause_3, negated_conjecture)
$\neg r$ cnf(clause_4, negated_conjecture)

LCL230-3.p Principia Mathematica 2.85
include('Axioms/LCL004-0.ax')
¬ theorem(implies(implies(or(p, q), or(p, r)), or(p, implies(q, r)))) cnf(prove_this, negated_conjecture)

LCL230^4.p Principia Mathematica 2.85
include('Axioms/LCL010^0.ax')
p: \$i → \$o thf(p_type, type)
q: \$i → \$o thf(q_type, type)
r: \$i → \$o thf(r_type, type)
ivalid@(iimplies@(iimplies@(ior@(iatom@p)@(iatom@q))@(ior@(iatom@p)@(iatom@r)))@(ior@(iatom@p)@(iimplies@(iatom

LCL231-1.p Principia Mathematica 2.86
include('Axioms/LCL003-0.ax')
¬ theorem(or(not(or(not(or(not(p), q)), or(not(p), r))), or(not(p), or(not(q), r)))) cnf(prove_this, negated_conjecture)

LCL231-3.p Principia Mathematica 2.86
include('Axioms/LCL004-0.ax')
¬ theorem(implies(implies(implies(p, q), implies(p, r)), implies(p, implies(q, r)))) cnf(prove_this, negated_conjecture)

LCL234-1.p Principia Mathematica 3.2 and 3.12
include('Axioms/LCL003-0.ax')
¬ theorem(or(not(p), or(not(q), not(or(not(p), not(q)))))) cnf(prove_this, negated_conjecture)

LCL234-3.p Principia Mathematica 3.2
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
¬ theorem(implies(p, implies(q, and(p, q)))) cnf(prove_this, negated_conjecture)

LCL235-1.p Principia Mathematica 3.13
include('Axioms/LCL003-0.ax')
¬ theorem(or(not(not(not(or(not(p), not(q))))), or(not(p), not(q)))) cnf(prove_this, negated_conjecture)

LCL235-3.p Principia Mathematica 3.13
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
¬ theorem(implies(not(and(p, q)), or(not(p), not(q)))) cnf(prove_this, negated_conjecture)

LCL236-1.p Principia Mathematica 3.14
include('Axioms/LCL003-0.ax')
¬ theorem(or(not(or(not(p), not(q))), not(not(or(not(p), not(q)))))) cnf(prove_this, negated_conjecture)

LCL236-3.p Principia Mathematica 3.14
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
¬ theorem(implies(or(not(p), not(q)), not(and(p, q)))) cnf(prove_this, negated_conjecture)

LCL237-1.p Principia Mathematica 3.21
include('Axioms/LCL003-0.ax')
¬ theorem(or(not(q), or(not(p), not(or(not(p), not(q)))))) cnf(prove_this, negated_conjecture)

LCL237-3.p Principia Mathematica 3.21
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
¬ theorem(implies(q, implies(p, and(p, q)))) cnf(prove_this, negated_conjecture)

LCL238-1.p Principia Mathematica 3.22

include('Axioms/LCL003-0.ax')
¬ theorem(or(not(not(or(not(p), not(q)))), not(or(not(q), not(p)))))   cnf(prove_this, negated_conjecture)

LCL238-3.p Principia Mathematica 3.22
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
¬ theorem(implies(and(p, q), and(q, p)))   cnf(prove_this, negated_conjecture)

LCL239-1.p Principia Mathematica 3.24
include('Axioms/LCL003-0.ax')
¬ theorem(not(not(or(not(p), not(not(p))))))   cnf(prove_this, negated_conjecture)

LCL239-3.p Principia Mathematica 3.24
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
¬ theorem(not(and(p, not(p))))   cnf(prove_this, negated_conjecture)

LCL240-1.p Principia Mathematica 3.26
include('Axioms/LCL003-0.ax')
¬ theorem(or(not(not(or(not(p), not(q)))), p))   cnf(prove_this, negated_conjecture)

LCL240-3.p Principia Mathematica 3.26
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
¬ theorem(implies(and(p, q), p))   cnf(prove_this, negated_conjecture)

LCL241-1.p Principia Mathematica 3.27
include('Axioms/LCL003-0.ax')
¬ theorem(or(not(not(or(not(p), not(q)))), q))   cnf(prove_this, negated_conjecture)

LCL241-3.p Principia Mathematica 3.27
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
¬ theorem(implies(and(p, q), q))   cnf(prove_this, negated_conjecture)

LCL242-1.p Principia Mathematica 3.3
include('Axioms/LCL003-0.ax')
¬ theorem(or(not(or(not(not(or(not(p), not(q)))), r)), or(not(p), or(not(q), r))))   cnf(prove_this, negated_conjecture)

LCL242-3.p Principia Mathematica 3.3
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
¬ theorem(implies(implies(and(p, q), r), implies(p, implies(q, r))))   cnf(prove_this, negated_conjecture)

LCL243-1.p Principia Mathematica 3.31
include('Axioms/LCL003-0.ax')
¬ theorem(or(not(or(not(p), or(not(q), r))), or(not(not(or(not(p), not(q)))), r)))   cnf(prove_this, negated_conjecture)

LCL243-3.p Principia Mathematica 3.31
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
¬ theorem(implies(implies(p, implies(q, r)), implies(and(p, q), r)))   cnf(prove_this, negated_conjecture)

LCL244-1.p Principia Mathematica 3.33
include('Axioms/LCL003-0.ax')
¬ theorem(or(not(not(or(not(or(not(p), q)), not(or(not(q), r))))), or(not(p), r)))   cnf(prove_this, negated_conjecture)

LCL245-1.p Principia Mathematica 3.34
include('Axioms/LCL003-0.ax')
¬ theorem(or(not(not(or(not(or(not(q), r)), not(or(not(p), q))))), or(not(p), r)))   cnf(prove_this, negated_conjecture)

LCL245-3.p Principia Mathematica 3.34
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
¬ theorem(implies(and(implies(q, r), implies(p, q)), implies(p, r)))   cnf(prove_this, negated_conjecture)

LCL246-1.p Principia Mathematica 3.35
include('Axioms/LCL003-0.ax')
¬ theorem(or(not(not(or(not(p), not(or(not(p), q))))), q))   cnf(prove_this, negated_conjecture)

LCL246-3.p Principia Mathematica 3.35
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
$\neg$ theorem(implies(and(p, implies(p, q)), q)) cnf(prove_this, negated_conjecture)

LCL247-1.p Principia Mathematica 3.37
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(or(not(not(or(not(p), not(q)))), r)), or(not(not(or(not(p), not(not(r))))), not(q)))) cnf(prove_this, negate

LCL247-3.p Principia Mathematica 3.37
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
$\neg$ theorem(implies(implies(and(p, q), r), implies(and(p, not(r)), not(q)))) cnf(prove_this, negated_conjecture)

LCL248-1.p Principia Mathematica 3.4
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(not(or(not(p), not(q)))), or(not(p), q))) cnf(prove_this, negated_conjecture)

LCL248-3.p Principia Mathematica 3.4
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
$\neg$ theorem(implies(and(p, q), implies(p, q))) cnf(prove_this, negated_conjecture)

LCL249-1.p Principia Mathematica 3.41
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(or(not(p), r)), or(not(not(or(not(p), not(q)))), r))) cnf(prove_this, negated_conjecture)

LCL249-3.p Principia Mathematica 3.41
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
$\neg$ theorem(implies(implies(p, r), implies(and(p, q), r))) cnf(prove_this, negated_conjecture)

LCL250-1.p Principia Mathematica 3.42
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(or(not(q), r)), or(not(not(or(not(p), not(q)))), r))) cnf(prove_this, negated_conjecture)

LCL250-3.p Principia Mathematica 3.42
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
$\neg$ theorem(implies(implies(q, r), implies(and(p, q), r))) cnf(prove_this, negated_conjecture)

LCL251-1.p Principia Mathematica 3.43
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(not(or(not(or(not(p), q)), not(or(not(p), r))))), or(not(p), not(or(not(q), not(r)))))) cnf(prove_this, negat

LCL251-3.p Principia Mathematica 3.43
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
$\neg$ theorem(implies(and(implies(p, q), implies(p, r)), implies(p, and(q, r)))) cnf(prove_this, negated_conjecture)

LCL252-1.p Principia Mathematica 3.44
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(not(or(not(or(not(q), p)), not(or(not(r), p))))), or(not(or(q, r)), p))) cnf(prove_this, negated_conjecture)

LCL252-3.p Principia Mathematica 3.44
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
$\neg$ theorem(implies(and(implies(q, p), implies(r, p)), implies(or(q, r), p))) cnf(prove_this, negated_conjecture)

LCL253-1.p Principia Mathematica 3.45
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(or(not(p), q)), or(not(not(or(not(p), not(r)))), not(or(not(q), not(r)))))) cnf(prove_this, negated_conjectu

LCL253-3.p Principia Mathematica 3.45
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
$\neg$ theorem(implies(implies(p, q), implies(and(p, r), and(q, r)))) cnf(prove_this, negated_conjecture)

LCL254-1.p Principia Mathematica 3.47

include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(not(or(not(or(not(p), r)), not(or(not(q), s))))), or(not(not(or(not(p), not(q)))), not(or(not(r), not(s))))))

LCL254-3.p Principia Mathematica 3.47
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
$\neg$ theorem(implies(and(implies(p, r), implies(q, s)), implies(and(p, q), and(r, s)))) cnf(prove_this, negated_conjecture)

LCL255-1.p Principia Mathematica 3.48
include('Axioms/LCL003-0.ax')
$\neg$ theorem(or(not(not(or(not(or(not(p), r)), not(or(not(q), s))))), or(not(or(p, q)), or(r, s)))) cnf(prove_this, negated_conjec

LCL255-3.p Principia Mathematica 3.48
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
$\neg$ theorem(implies(and(implies(p, r), implies(q, s)), implies(or(p, q), or(r, s)))) cnf(prove_this, negated_conjecture)

LCL256-1.p A formula that can be derived from the Lukasiewicz system
Axiomatisations of the Implication/Negation 2 valued sentential calculus are CN-1,CN-2,CN-3 by Lukasiewicz, CN-18,CN-21,CN-35,CN-39,CN-39,CN-40,CN-46 by Frege, CN-3,CN-18,CN-21,CN-22,CN-30,CN-54 by Hilbert, CN-18, CN-35,CN-49 by Church, CN-19,CN-37,CN-59 by Lukasiewicz, CN-19,CN-37,CN-60 by Wos, and the single Meredith axiom. Show that not(not(p → p)) can be derived from the short Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z)))) cnf(cn_1, axiom)
is_a_theorem(implies(implies(not(x), x), x)) cnf(cn_2, axiom)
is_a_theorem(implies(x, implies(not(x), y))) cnf(cn_3, axiom)
$\neg$ is_a_theorem(not(not(implies(p, p)))) cnf(prove_not_not_implies, negated_conjecture)

LCL257-1.p XHN depends on YQL
Show that XHN can be derived from the single Lukasiewicz axiom YQL.
(is_a_theorem(equivalent(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(equivalent(equivalent(x, y), equivalent(equivalent(z, y), equivalent(x, z)))) cnf(yql, axiom)
$\neg$ is_a_theorem(equivalent(x, equivalent(equivalent(y, z), equivalent(equivalent(z, x), y)))) cnf(prove_xhn, negated_conjectu

LCL258-3.p Principia Mathematica 2.27
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(p, implies(implies(p, q), q))) cnf(prove_this, negated_conjecture)

LCL259-3.p Principia Mathematica 2.43
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(implies(p, implies(p, q)), implies(p, q))) cnf(prove_this, negated_conjecture)

LCL260-3.p Principia Mathematica 2.63
include('Axioms/LCL004-0.ax')
$\neg$ theorem(implies(or(p, q), implies(or(not(p), q), q))) cnf(prove_this, negated_conjecture)

LCL261-3.p Principia Mathematica 3.12
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
$\neg$ theorem(or(not(p), or(not(q), and(p, q)))) cnf(prove_this, negated_conjecture)

LCL262-3.p Principia Mathematica 4.10
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(implies(p, q), implies(not(q), not(p)))) cnf(prove_this, negated_conjecture)

LCL263-3.p Principia Mathematica 4.11
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(equivalent(p, q), equivalent(not(p), not(q)))) cnf(prove_this, negated_conjecture)

LCL264-3.p Principia Mathematica 4.12
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')

$\neg$ theorem(equivalent(equivalent(p, q), equivalent(not(p), not(q)))) cnf(prove_this, negated_conjecture)

LCL265-3.p Principia Mathematica 4.13
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(p, not(not(p)))) cnf(prove_this, negated_conjecture)

LCL266-3.p Principia Mathematica 4.14
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(and(p, implies(q, r)), and(p, implies(not(r), not(q))))) cnf(prove_this, negated_conjecture)

LCL267-3.p Principia Mathematica 4.15
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(and(p, implies(q, not(r))), and(q, implies(r, not(p))))) cnf(prove_this, negated_conjecture)

LCL268-3.p Principia Mathematica 4.2
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(p, p)) cnf(prove_this, negated_conjecture)

LCL269-3.p Principia Mathematica 4.21
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(equivalent(p, q), equivalent(q, p))) cnf(prove_this, negated_conjecture)

LCL270-3.p Principia Mathematica 4.22
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(implies(and(equivalent(p, q), equivalent(q, r)), equivalent(p, r))) cnf(prove_this, negated_conjecture)

LCL271-3.p Principia Mathematica 4.24
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(p, and(p, p))) cnf(prove_this, negated_conjecture)

LCL272-3.p Principia Mathematica 4.25
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(p, or(p, p))) cnf(prove_this, negated_conjecture)

LCL273-3.p Principia Mathematica 4.3
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(and(p, q), and(q, p))) cnf(prove_this, negated_conjecture)

LCL274-3.p Principia Mathematica 4.31
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(or(p, q), or(q, p))) cnf(prove_this, negated_conjecture)

LCL275-3.p Principia Mathematica 4.32
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')

$\neg$ theorem(equivalent(and(r, and(p, q)), and(p, and(q, r)))) cnf(prove_this, negated_conjecture)

LCL276-3.p Principia Mathematica 4.33
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(or(r, or(p, q)), or(p, or(q, r)))) cnf(prove_this, negated_conjecture)

LCL277-3.p Principia Mathematica 4.36
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(implies(equivalent(p, q), equivalent(and(p, r), and(q, r)))) cnf(prove_this, negated_conjecture)

LCL278-3.p Principia Mathematica 4.37
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(implies(equivalent(p, q), equivalent(or(p, r), or(q, r)))) cnf(prove_this, negated_conjecture)

LCL279-3.p Principia Mathematica 4.38
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(implies(and(equivalent(p, r), equivalent(q, s)), equivalent(and(p, q), and(r, s)))) cnf(prove_this, negated_conjectu

LCL280-3.p Principia Mathematica 4.39
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(implies(and(equivalent(p, r), equivalent(q, s)), equivalent(or(p, r), or(r, s)))) cnf(prove_this, negated_conjecture)

LCL281-3.p Principia Mathematica 4.4
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(and(p, or(q, r)), or(and(p, q), and(p, r)))) cnf(prove_this, negated_conjecture)

LCL282-3.p Principia Mathematica 4.41
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(or(p, and(q, r)), and(or(p, q), or(p, r)))) cnf(prove_this, negated_conjecture)

LCL283-3.p Principia Mathematica 4.42
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(p, or(and(p, q), and(p, not(q))))) cnf(prove_this, negated_conjecture)

LCL284-3.p Principia Mathematica 4.43
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(p, and(or(p, q), or(p, not(q))))) cnf(prove_this, negated_conjecture)

LCL285-3.p Principia Mathematica 4.44
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(p, or(p, and(p, q)))) cnf(prove_this, negated_conjecture)

LCL286-3.p Principia Mathematica 4.45
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')

$\neg$ theorem(equivalent(p, and(p, or(p, q)))) cnf(prove_this, negated_conjecture)

LCL287-3.p Principia Mathematica 4.5
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(and(p, q), not(or(not(p), not(q))))) cnf(prove_this, negated_conjecture)

LCL288-3.p Principia Mathematica 4.52
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(and(p, not(q)), or(not(p), not(q)))) cnf(prove_this, negated_conjecture)

LCL289-3.p Principia Mathematica 4.53
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(not(and(p, not(q))), or(not(p), q))) cnf(prove_this, negated_conjecture)

LCL290-3.p Principia Mathematica 4.54
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(not(and(p, q)), not(or(p, not(q))))) cnf(prove_this, negated_conjecture)

LCL291-3.p Principia Mathematica 4.55
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(not(not(and(p, q))), or(not(p), q))) cnf(prove_this, negated_conjecture)

LCL292-3.p Principia Mathematica 4.56
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(and(not(p), not(q)), or(p, q))) cnf(prove_this, negated_conjecture)

LCL293-3.p Principia Mathematica 4.57
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(not(and(not(p), not(q))), or(p, q))) cnf(prove_this, negated_conjecture)

LCL294-3.p Principia Mathematica 4.6
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(implies(p, q), or(not(p), q))) cnf(prove_this, negated_conjecture)

LCL295-3.p Principia Mathematica 4.61
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(not(implies(p, q)), and(p, not(q)))) cnf(prove_this, negated_conjecture)

LCL296-3.p Principia Mathematica 4.62
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(implies(p, not(q)), or(not(p), not(q)))) cnf(prove_this, negated_conjecture)

LCL297-3.p Principia Mathematica 4.63
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')

$\neg$ theorem(equivalent(not(implies(p, not(q))), and(p, q))) cnf(prove_this, negated_conjecture)

LCL298-3.p Principia Mathematica 4.51
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(not(and(p, q)), or(not(p), not(q)))) cnf(prove_this, negated_conjecture)

LCL299-3.p Principia Mathematica 4.65
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(not(implies(not(p), q)), and(not(p), not(q)))) cnf(prove_this, negated_conjecture)

LCL300-3.p Principia Mathematica 4.66
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(implies(not(p), not(q)), or(p, not(q)))) cnf(prove_this, negated_conjecture)

LCL301-3.p Principia Mathematica 4.67
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(not(implies(not(p), not(q))), and(not(p), q))) cnf(prove_this, negated_conjecture)

LCL302-3.p Principia Mathematica 4.7
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(implies(p, q), implies(p, and(p, q)))) cnf(prove_this, negated_conjecture)

LCL303-3.p Principia Mathematica 4.71
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(implies(p, q), equivalent(q, or(p, q)))) cnf(prove_this, negated_conjecture)

LCL304-3.p Principia Mathematica 4.72
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(implies(p, q), equivalent(q, or(p, q)))) cnf(prove_this, negated_conjecture)

LCL305-3.p Principia Mathematica 4.73
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(implies(q, equivalent(p, and(p, q)))) cnf(prove_this, negated_conjecture)

LCL306-3.p Principia Mathematica 4.74
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(implies(not(p), equivalent(q, or(p, q)))) cnf(prove_this, negated_conjecture)

LCL307-3.p Principia Mathematica 4.76
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(and(implies(p, q), implies(p, r)), implies(p, and(q, r)))) cnf(prove_this, negated_conjecture)

LCL308-3.p Principia Mathematica 4.77
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')

¬ theorem(equivalent(and(implies(q, p), implies(r, p)), implies(or(q, r), p))) cnf(prove_this, negated_conjecture)

LCL309-3.p Principia Mathematica 4.78
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
¬ theorem(equivalent(or(implies(p, q), implies(p, r)), implies(p, or(q, r)))) cnf(prove_this, negated_conjecture)

LCL310-3.p Principia Mathematica 4.79
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
¬ theorem(equivalent(or(implies(q, p), implies(r, p)), implies(and(q, r), p))) cnf(prove_this, negated_conjecture)

LCL311-3.p Principia Mathematica 4.8
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
¬ theorem(equivalent(implies(p, not(p)), not(p))) cnf(prove_this, negated_conjecture)

LCL312-3.p Principia Mathematica 4.81
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
¬ theorem(equivalent(implies(not(p), p), p)) cnf(prove_this, negated_conjecture)

LCL313-3.p Principia Mathematica 4.82
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
¬ theorem(equivalent(and(implies(p, q), implies(p, not(q))), not(p))) cnf(prove_this, negated_conjecture)

LCL314-3.p Principia Mathematica 4.83
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
¬ theorem(equivalent(and(implies(p, q), implies(not(p), q)), q)) cnf(prove_this, negated_conjecture)

LCL315-3.p Principia Mathematica 4.84
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
¬ theorem(implies(equivalent(p, q), equivalent(implies(p, r), implies(q, r)))) cnf(prove_this, negated_conjecture)

LCL316-3.p Principia Mathematica 4.85
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
¬ theorem(implies(equivalent(p, q), equivalent(implies(r, p), implies(r, q)))) cnf(prove_this, negated_conjecture)

LCL317-3.p Principia Mathematica 4.86
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
¬ theorem(implies(equivalent(p, q), equivalent(equivalent(p, r), equivalent(q, r)))) cnf(prove_this, negated_conjecture)

LCL318-3.p Principia Mathematica 4.87
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
¬ theorem(equivalent(implies(and(p, q), r), equivalent(implies(p, implies(q, r)), implies(q, equivalent(implies(p, r), implies(and(

LCL319-3.p Principia Mathematica 5.1
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')

¬ theorem(implies(and(p, q), equivalent(p, q))) cnf(prove_this, negated_conjecture)

LCL320-3.p Principia Mathematica 5.11
include('Axioms/LCL004-0.ax')
¬ theorem(or(implies(p, q), implies(not(p), q))) cnf(prove_this, negated_conjecture)

LCL321-3.p Principia Mathematica 5.12
include('Axioms/LCL004-0.ax')
¬ theorem(or(implies(p, q), implies(p, not(q)))) cnf(prove_this, negated_conjecture)

LCL322-3.p Principia Mathematica 5.13
include('Axioms/LCL004-0.ax')
¬ theorem(or(implies(p, q), implies(q, p))) cnf(prove_this, negated_conjecture)

LCL323-3.p Principia Mathematica 5.14
include('Axioms/LCL004-0.ax')
¬ theorem(or(implies(p, q), implies(q, r))) cnf(prove_this, negated_conjecture)

LCL324-3.p Principia Mathematica 5.16
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
¬ theorem(not(and(equivalent(p, q), equivalent(p, not(q))))) cnf(prove_this, negated_conjecture)

LCL325-3.p Principia Mathematica 5.17
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
¬ theorem(equivalent(and(or(p, q), not(and(p, q))), equivalent(p, not(q)))) cnf(prove_this, negated_conjecture)

LCL326-3.p Principia Mathematica 5.19
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
¬ theorem(not(equivalent(p, not(p)))) cnf(prove_this, negated_conjecture)

LCL327-3.p Principia Mathematica 5.21
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
¬ theorem(implies(and(not(p), not(q)), equivalent(p, q))) cnf(prove_this, negated_conjecture)

LCL328-3.p Principia Mathematica 5.23
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
¬ theorem(equivalent(equivalent(p, q), or(and(p, q), and(not(p), not(q))))) cnf(prove_this, negated_conjecture)

LCL329-3.p Principia Mathematica 5.24
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
¬ theorem(equivalent(not(or(and(p, q), and(not(p), not(q)))), or(and(p, not(q)), and(q, not(p))))) cnf(prove_this, negated_co

LCL330-3.p Principia Mathematica 5.25
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
¬ theorem(equivalent(or(p, q), implies(implies(p, q), q))) cnf(prove_this, negated_conjecture)

LCL331-3.p Principia Mathematica 5.3
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
¬ theorem(equivalent(implies(and(p, q), r), implies(and(p, q), and(p, r)))) cnf(prove_this, negated_conjecture)

LCL332-3.p Principia Mathematica 5.31

include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
$\neg$ theorem(implies(and(r, implies(p, q)), implies(p, and(q, r)))) cnf(prove_this, negated_conjecture)

LCL333-3.p Principia Mathematica 5.32
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(implies(p, equivalent(q, r)), equivalent(and(p, q), and(p, r)))) cnf(prove_this, negated_conjecture)

LCL334-3.p Principia Mathematica 5.33
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(and(p, implies(q, r)), and(p, implies(and(p, q), r)))) cnf(prove_this, negated_conjecture)

LCL335-3.p Principia Mathematica 5.35
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(implies(and(implies(p, q), implies(q, r)), implies(p, equivalent(q, r)))) cnf(prove_this, negated_conjecture)

LCL336-3.p Principia Mathematica 5.36
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(and(p, equivalent(p, q)), and(q, equivalent(p, q)))) cnf(prove_this, negated_conjecture)

LCL337-3.p Principia Mathematica 5.4
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(implies(p, implies(p, q)), implies(p, q))) cnf(prove_this, negated_conjecture)

LCL338-3.p Principia Mathematica 5.41
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(implies(implies(p, q), implies(q, r)), implies(p, implies(q, r)))) cnf(prove_this, negated_conjecture)

LCL339-3.p Principia Mathematica 5.42
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(implies(p, implies(q, r)), implies(p, implies(q, and(p, r))))) cnf(prove_this, negated_conjecture)

LCL340-3.p Principia Mathematica 5.44
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(implies(implies(p, q), equivalent(implies(p, r), implies(p, and(q, r))))) cnf(prove_this, negated_conjecture)

LCL341-3.p Principia Mathematica 5.5
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(implies(p, equivalent(implies(p, q), q))) cnf(prove_this, negated_conjecture)

LCL342-3.p Principia Mathematica 5.501
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(implies(p, equivalent(q, equivalent(p, q)))) cnf(prove_this, negated_conjecture)

LCL343-3.p Principia Mathematica 5.53
include('Axioms/LCL004-0.ax')

include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(implies(or(or(p,q), r), s), and(and(implies(p,s), implies(q,s)), implies(r,s)))) cnf(prove_this, negated

LCL344-3.p Principia Mathematica 5.54
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(or(equivalent(and(p,q), p), equivalent(and(p,q), q))) cnf(prove_this, negated_conjecture)

LCL345-3.p Principia Mathematica 5.55
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(or(equivalent(or(p,q), p), equivalent(or(p,q), q))) cnf(prove_this, negated_conjecture)

LCL346-3.p Principia Mathematica 5.6
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(implies(and(p, not(q)), r), implies(p, or(q,r)))) cnf(prove_this, negated_conjecture)

LCL347-3.p Principia Mathematica 5.61
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(and(or(p,q), not(q)), and(p, not(q)))) cnf(prove_this, negated_conjecture)

LCL348-3.p Principia Mathematica 5.62
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(or(and(p,q), not(q)), or(p, not(q)))) cnf(prove_this, negated_conjecture)

LCL349-3.p Principia Mathematica 5.63
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(or(p,q), or(p, and(not(p), q)))) cnf(prove_this, negated_conjecture)

LCL350-3.p Principia Mathematica 5.7
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(equivalent(or(p,r), or(q,r)), or(r, equivalent(p,q)))) cnf(prove_this, negated_conjecture)

LCL351-3.p Principia Mathematica 5.71
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(implies(implies(q, not(r)), equivalent(and(or(p,q), r), and(p,r)))) cnf(prove_this, negated_conjecture)

LCL352-3.p Principia Mathematica 5.74
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(equivalent(implies(p, equivalent(q,r)), equivalent(implies(p,q), implies(p,r)))) cnf(prove_this, negated_conjectu

LCL353-3.p Principia Mathematica 5.75
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')
$\neg$ theorem(implies(equivalent(and(implies(r, not(q)), p), or(q,r)), equivalent(and(p, not(q)), r))) cnf(prove_this, negated_co

LCL354+1.p Independence of an Axiom for Temporal Intervals

Shows that the 5th axiom of temporal intervals is not dependant on the first three by building a model of the first three and the negation of the 5th.
$\forall p, q, r, s$: ((meets(p, q) and meets(p, s) and meets(r, q)) $\Rightarrow$ meets(r, s)) fof(m_1, axiom)
$\forall p, q, r, s$: ((meets(p, q) and meets(r, s)) $\Rightarrow$ meets(p, s) $<>$ $\exists t$: (meets(p, t) and meets(t, s)) $<>$ $\exists t$: (meets(r, t) and meets($t,$
$\forall p$: $\exists q, r$: (meets(q, p) and meets(p, r)) fof(m_3, axiom)
$\neg\,\forall p, q$: (meets(p, q) $\Rightarrow$ $\exists r, s, t$: (meets(r, p) and meets(q, s) and meets(r, t) and meets(t, s))) fof(not_m$_5$, axiom)

LCL355-1.p CN-04 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-04 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z)))) cnf(cn$_1$, axiom)
is_a_theorem(implies(implies(not(x), x), x)) cnf(cn$_2$, axiom)
is_a_theorem(implies(x, implies(not(x), y))) cnf(cn$_3$, axiom)
$\neg$ is_a_theorem(implies(implies(implies(implies(x, y), implies(z, y)), u), implies(implies(z, x), u))) cnf(prove_cn$_{04}$, negated_c

LCL356-1.p CN-05 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-05 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z)))) cnf(cn$_1$, axiom)
is_a_theorem(implies(implies(not(x), x), x)) cnf(cn$_2$, axiom)
is_a_theorem(implies(x, implies(not(x), y))) cnf(cn$_3$, axiom)
$\neg$ is_a_theorem(implies(implies(x, implies(y, z)), implies(implies(u, y), implies(x, implies(u, z))))) cnf(prove_cn$_{05}$, negated_c

LCL357-1.p CN-06 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-06 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z)))) cnf(cn$_1$, axiom)
is_a_theorem(implies(implies(not(x), x), x)) cnf(cn$_2$, axiom)
is_a_theorem(implies(x, implies(not(x), y))) cnf(cn$_3$, axiom)
$\neg$ is_a_theorem(implies(implies(x, y), implies(implies(implies(x, z), u), implies(implies(y, z), u)))) cnf(prove_cn$_{06}$, negated_c

LCL358-1.p CN-07 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-07 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z)))) cnf(cn$_1$, axiom)
is_a_theorem(implies(implies(not(x), x), x)) cnf(cn$_2$, axiom)
is_a_theorem(implies(x, implies(not(x), y))) cnf(cn$_3$, axiom)
$\neg$ is_a_theorem(implies(implies(x, implies(implies(y, z), u)), implies(implies(y, v), implies(x, implies(implies(v, z), u))))) cnf

LCL359-1.p CN-08 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-08 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z)))) cnf(cn$_1$, axiom)
is_a_theorem(implies(implies(not(x), x), x)) cnf(cn$_2$, axiom)
is_a_theorem(implies(x, implies(not(x), y))) cnf(cn$_3$, axiom)
$\neg$ is_a_theorem(implies(implies(x, y), implies(implies(z, x), implies(implies(y, u), implies(z, u))))) cnf(prove_cn$_{08}$, negated_c

LCL360-1.p CN-09 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-09 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z)))) cnf(cn$_1$, axiom)
is_a_theorem(implies(implies(not(x), x), x)) cnf(cn$_2$, axiom)
is_a_theorem(implies(x, implies(not(x), y))) cnf(cn$_3$, axiom)
$\neg$ is_a_theorem(implies(implies(implies(not(x), y), z), implies(x, z))) cnf(prove_cn$_{09}$, negated_conjecture)

LCL361-1.p CN-10 depends on the Lukasiewicz system

An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-10 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z))))   cnf(cn_1, axiom)
is_a_theorem(implies(implies(not(x), x), x))   cnf(cn_2, axiom)
is_a_theorem(implies(x, implies(not(x), y)))   cnf(cn_3, axiom)
¬ is_a_theorem(implies(x, implies(implies(implies(not(x), x), x), implies(implies(y, x), x))))   cnf(prove_cn_{10}, negated_conject

LCL362-1.p CN-11 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-11 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z))))   cnf(cn_1, axiom)
is_a_theorem(implies(implies(not(x), x), x))   cnf(cn_2, axiom)
is_a_theorem(implies(x, implies(not(x), y)))   cnf(cn_3, axiom)
¬ is_a_theorem(implies(implies(x, implies(implies(not(y), y), y)), implies(implies(not(y), y), y)))   cnf(prove_cn_{11}, negated_co

LCL363-1.p CN-12 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-12 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z))))   cnf(cn_1, axiom)
is_a_theorem(implies(implies(not(x), x), x))   cnf(cn_2, axiom)
is_a_theorem(implies(x, implies(not(x), y)))   cnf(cn_3, axiom)
¬ is_a_theorem(implies(x, implies(implies(not(y), y), y)))   cnf(prove_cn_{12}, negated_conjecture)

LCL364-1.p CN-13 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-13 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z))))   cnf(cn_1, axiom)
is_a_theorem(implies(implies(not(x), x), x))   cnf(cn_2, axiom)
is_a_theorem(implies(x, implies(not(x), y)))   cnf(cn_3, axiom)
¬ is_a_theorem(implies(implies(not(x), y), implies(z, implies(implies(y, x), x))))   cnf(prove_cn_{13}, negated_conjecture)

LCL365-1.p CN-14 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-14 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z))))   cnf(cn_1, axiom)
is_a_theorem(implies(implies(not(x), x), x))   cnf(cn_2, axiom)
is_a_theorem(implies(x, implies(not(x), y)))   cnf(cn_3, axiom)
¬ is_a_theorem(implies(implies(implies(x, implies(implies(y, z), z)), u), implies(implies(not(z), y), u)))   cnf(prove_cn_{14}, nega

LCL366-1.p CN-15 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-15 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z))))   cnf(cn_1, axiom)
is_a_theorem(implies(implies(not(x), x), x))   cnf(cn_2, axiom)
is_a_theorem(implies(x, implies(not(x), y)))   cnf(cn_3, axiom)
¬ is_a_theorem(implies(implies(not(x), y), implies(implies(y, x), x)))   cnf(prove_cn_{15}, negated_conjecture)

LCL367-1.p CN-17 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-17 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z))))   cnf(cn_1, axiom)
is_a_theorem(implies(implies(not(x), x), x))   cnf(cn_2, axiom)
is_a_theorem(implies(x, implies(not(x), y)))   cnf(cn_3, axiom)
¬ is_a_theorem(implies(x, implies(implies(y, x), x)))   cnf(prove_cn_{17}, negated_conjecture)

LCL368-1.p CN-23 depends on the Lukasiewicz system

An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-23 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z))))   cnf(cn$_1$, axiom)
is_a_theorem(implies(implies(not(x), x), x))   cnf(cn$_2$, axiom)
is_a_theorem(implies(x, implies(not(x), y)))   cnf(cn$_3$, axiom)
¬ is_a_theorem(implies(implies(implies(x, implies(y, z)), u), implies(implies(y, implies(x, z)), u)))   cnf(prove_cn$_{23}$, negated_c

LCL369-1.p CN-25 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-25 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z))))   cnf(cn$_1$, axiom)
is_a_theorem(implies(implies(not(x), x), x))   cnf(cn$_2$, axiom)
is_a_theorem(implies(x, implies(not(x), y)))   cnf(cn$_3$, axiom)
¬ is_a_theorem(implies(implies(implies(x, y), z), implies(implies(x, u), implies(implies(u, y), z))))   cnf(prove_cn$_{25}$, negated_c

LCL370-1.p CN-26 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-26 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z))))   cnf(cn$_1$, axiom)
is_a_theorem(implies(implies(not(x), x), x))   cnf(cn$_2$, axiom)
is_a_theorem(implies(x, implies(not(x), y)))   cnf(cn$_3$, axiom)
¬ is_a_theorem(implies(implies(implies(x, y), z), implies(implies(z, x), x)))   cnf(prove_cn$_{26}$, negated_conjecture)

LCL371-1.p CN-27 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-27 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z))))   cnf(cn$_1$, axiom)
is_a_theorem(implies(implies(not(x), x), x))   cnf(cn$_2$, axiom)
is_a_theorem(implies(x, implies(not(x), y)))   cnf(cn$_3$, axiom)
¬ is_a_theorem(implies(implies(implies(x, y), y), implies(implies(y, x), x)))   cnf(prove_cn$_{27}$, negated_conjecture)

LCL372-1.p CN-28 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-28 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z))))   cnf(cn$_1$, axiom)
is_a_theorem(implies(implies(not(x), x), x))   cnf(cn$_2$, axiom)
is_a_theorem(implies(x, implies(not(x), y)))   cnf(cn$_3$, axiom)
¬ is_a_theorem(implies(implies(implies(implies(x, y), y), z), implies(implies(implies(y, u), x), z)))   cnf(prove_cn$_{28}$, negated_c

LCL373-1.p CN-29 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-29 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z))))   cnf(cn$_1$, axiom)
is_a_theorem(implies(implies(not(x), x), x))   cnf(cn$_2$, axiom)
is_a_theorem(implies(x, implies(not(x), y)))   cnf(cn$_3$, axiom)
¬ is_a_theorem(implies(implies(implies(x, y), z), implies(implies(x, z), z)))   cnf(prove_cn$_{29}$, negated_conjecture)

LCL374-1.p CN-31 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-31 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z))))   cnf(cn$_1$, axiom)
is_a_theorem(implies(implies(not(x), x), x))   cnf(cn$_2$, axiom)
is_a_theorem(implies(x, implies(not(x), y)))   cnf(cn$_3$, axiom)
¬ is_a_theorem(implies(implies(x, y), implies(implies(implies(x, z), u), implies(implies(y, u), u))))   cnf(prove_cn$_{31}$, negated_c

LCL375-1.p CN-32 depends on the Lukasiewicz system

An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-32 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z)))) cnf(cn_1, axiom)
is_a_theorem(implies(implies(not(x), x), x)) cnf(cn_2, axiom)
is_a_theorem(implies(x, implies(not(x), y))) cnf(cn_3, axiom)
¬ is_a_theorem(implies(implies(implies(x, y), z), implies(implies(x, u), implies(implies(u, z), z)))) cnf(prove_cn_{32}, negated_c

LCL376-1.p CN-33 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-33 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z)))) cnf(cn_1, axiom)
is_a_theorem(implies(implies(not(x), x), x)) cnf(cn_2, axiom)
is_a_theorem(implies(x, implies(not(x), y))) cnf(cn_3, axiom)
¬ is_a_theorem(implies(implies(x, y), implies(implies(y, implies(z, implies(x, u))), implies(z, implies(x, u))))) cnf(prove_cn3

LCL377-1.p CN-34 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-34 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z)))) cnf(cn_1, axiom)
is_a_theorem(implies(implies(not(x), x), x)) cnf(cn_2, axiom)
is_a_theorem(implies(x, implies(not(x), y))) cnf(cn_3, axiom)
¬ is_a_theorem(implies(implies(x, implies(y, implies(z, u))), implies(implies(z, x), implies(y, implies(z, u))))) cnf(prove_cn3

LCL378-1.p CN-36 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-36 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z)))) cnf(cn_1, axiom)
is_a_theorem(implies(implies(not(x), x), x)) cnf(cn_2, axiom)
is_a_theorem(implies(x, implies(not(x), y))) cnf(cn_3, axiom)
¬ is_a_theorem(implies(not(x), implies(x, y))) cnf(prove_cn_{36}, negated_conjecture)

LCL379-1.p CN-38 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-38 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z)))) cnf(cn_1, axiom)
is_a_theorem(implies(implies(not(x), x), x)) cnf(cn_2, axiom)
is_a_theorem(implies(x, implies(not(x), y))) cnf(cn_3, axiom)
¬ is_a_theorem(implies(implies(x, not(x)), not(x))) cnf(prove_cn_{38}, negated_conjecture)

LCL380-1.p CN-41 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-41 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z)))) cnf(cn_1, axiom)
is_a_theorem(implies(implies(not(x), x), x)) cnf(cn_2, axiom)
is_a_theorem(implies(x, implies(not(x), y))) cnf(cn_3, axiom)
¬ is_a_theorem(implies(implies(x, y), implies(not(not(x)), y))) cnf(prove_cn_{41}, negated_conjecture)

LCL381-1.p CN-42 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-42 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z)))) cnf(cn_1, axiom)
is_a_theorem(implies(implies(not(x), x), x)) cnf(cn_2, axiom)
is_a_theorem(implies(x, implies(not(x), y))) cnf(cn_3, axiom)
¬ is_a_theorem(implies(implies(implies(not(not(x)), y), z), implies(implies(x, y), z))) cnf(prove_cn_{42}, negated_conjecture)

LCL382-1.p CN-43 depends on the Lukasiewicz system

An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-43 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z))))   cnf(cn$_1$, axiom)
is_a_theorem(implies(implies(not(x), x), x))   cnf(cn$_2$, axiom)
is_a_theorem(implies(x, implies(not(x), y)))   cnf(cn$_3$, axiom)
$\neg$ is_a_theorem(implies(implies(x, y), implies(implies(y, not(x)), not(x))))   cnf(prove_cn$_{43}$, negated_conjecture)

LCL383-1.p CN-44 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-44 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z))))   cnf(cn$_1$, axiom)
is_a_theorem(implies(implies(not(x), x), x))   cnf(cn$_2$, axiom)
is_a_theorem(implies(x, implies(not(x), y)))   cnf(cn$_3$, axiom)
$\neg$ is_a_theorem(implies(implies(x, implies(y, not(z))), implies(implies(z, y), implies(x, not(z)))))   cnf(prove_cn$_{44}$, negated_co

LCL384-1.p CN-45 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-45 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z))))   cnf(cn$_1$, axiom)
is_a_theorem(implies(implies(not(x), x), x))   cnf(cn$_2$, axiom)
is_a_theorem(implies(x, implies(not(x), y)))   cnf(cn$_3$, axiom)
$\neg$ is_a_theorem(implies(implies(x, implies(y, z)), implies(implies(not(z), y), implies(x, z))))   cnf(prove_cn$_{45}$, negated_conject

LCL385-1.p CN-47 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-47 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z))))   cnf(cn$_1$, axiom)
is_a_theorem(implies(implies(not(x), x), x))   cnf(cn$_2$, axiom)
is_a_theorem(implies(x, implies(not(x), y)))   cnf(cn$_3$, axiom)
$\neg$ is_a_theorem(implies(implies(x, not(y)), implies(y, not(x))))   cnf(prove_cn$_{47}$, negated_conjecture)

LCL386-1.p CN-48 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-48 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z))))   cnf(cn$_1$, axiom)
is_a_theorem(implies(implies(not(x), x), x))   cnf(cn$_2$, axiom)
is_a_theorem(implies(x, implies(not(x), y)))   cnf(cn$_3$, axiom)
$\neg$ is_a_theorem(implies(implies(not(x), y), implies(not(y), x)))   cnf(prove_cn$_{48}$, negated_conjecture)

LCL387-1.p CN-50 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-50 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z))))   cnf(cn$_1$, axiom)
is_a_theorem(implies(implies(not(x), x), x))   cnf(cn$_2$, axiom)
is_a_theorem(implies(x, implies(not(x), y)))   cnf(cn$_3$, axiom)
$\neg$ is_a_theorem(implies(implies(implies(not(x), y), z), implies(implies(not(y), x), z)))   cnf(prove_cn$_{50}$, negated_conjecture)

LCL388-1.p CN-51 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-51 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z))))   cnf(cn$_1$, axiom)
is_a_theorem(implies(implies(not(x), x), x))   cnf(cn$_2$, axiom)
is_a_theorem(implies(x, implies(not(x), y)))   cnf(cn$_3$, axiom)
$\neg$ is_a_theorem(implies(implies(x, implies(y, z)), implies(x, implies(not(z), not(y)))))   cnf(prove_cn$_{51}$, negated_conjecture)

LCL389-1.p CN-52 depends on the Lukasiewicz system

An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-52 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z)))) cnf(cn$_1$, axiom)
is_a_theorem(implies(implies(not(x), x), x)) cnf(cn$_2$, axiom)
is_a_theorem(implies(x, implies(not(x), y))) cnf(cn$_3$, axiom)
$\neg$ is_a_theorem(implies(implies(x, implies(y, not(z))), implies(x, implies(z, not(y))))) cnf(prove_cn$_{52}$, negated_conjecture)

LCL390-1.p CN-53 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-53 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z)))) cnf(cn$_1$, axiom)
is_a_theorem(implies(implies(not(x), x), x)) cnf(cn$_2$, axiom)
is_a_theorem(implies(x, implies(not(x), y))) cnf(cn$_3$, axiom)
$\neg$ is_a_theorem(implies(implies(not(x), y), implies(implies(x, y), y))) cnf(prove_cn$_{53}$, negated_conjecture)

LCL391-1.p CN-55 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-55 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z)))) cnf(cn$_1$, axiom)
is_a_theorem(implies(implies(not(x), x), x)) cnf(cn$_2$, axiom)
is_a_theorem(implies(x, implies(not(x), y))) cnf(cn$_3$, axiom)
$\neg$ is_a_theorem(implies(implies(x, y), implies(implies(x, not(y)), not(x)))) cnf(prove_cn$_{55}$, negated_conjecture)

LCL392-1.p CN-56 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-56 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z)))) cnf(cn$_1$, axiom)
is_a_theorem(implies(implies(not(x), x), x)) cnf(cn$_2$, axiom)
is_a_theorem(implies(x, implies(not(x), y))) cnf(cn$_3$, axiom)
$\neg$ is_a_theorem(implies(implies(implies(implies(x, y), y), z), implies(implies(not(x), y), z))) cnf(prove_cn$_{56}$, negated_conject

LCL393-1.p CN-57 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-57 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z)))) cnf(cn$_1$, axiom)
is_a_theorem(implies(implies(not(x), x), x)) cnf(cn$_2$, axiom)
is_a_theorem(implies(x, implies(not(x), y))) cnf(cn$_3$, axiom)
$\neg$ is_a_theorem(implies(implies(not(x), y), implies(implies(x, z), implies(implies(z, y), y)))) cnf(prove_cn$_{57}$, negated_conject

LCL394-1.p CN-58 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-58 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z)))) cnf(cn$_1$, axiom)
is_a_theorem(implies(implies(not(x), x), x)) cnf(cn$_2$, axiom)
is_a_theorem(implies(x, implies(not(x), y))) cnf(cn$_3$, axiom)
$\neg$ is_a_theorem(implies(implies(implies(implies(x, y), implies(implies(y, z), z)), u), implies(implies(not(x), z), u))) cnf(prove

LCL395-1.p CN-61 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-61 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z)))) cnf(cn$_1$, axiom)
is_a_theorem(implies(implies(not(x), x), x)) cnf(cn$_2$, axiom)
is_a_theorem(implies(x, implies(not(x), y))) cnf(cn$_3$, axiom)
$\neg$ is_a_theorem(implies(implies(x, y), implies(implies(z, y), implies(implies(not(x), z), y)))) cnf(prove_cn$_{61}$, negated_conject

LCL396-1.p CN-62 depends on the Lukasiewicz system

An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-62 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z)))) cnf(cn_1, axiom)
is_a_theorem(implies(implies(not(x), x), x)) cnf(cn_2, axiom)
is_a_theorem(implies(x, implies(not(x), y))) cnf(cn_3, axiom)
$\neg$ is_a_theorem(implies(implies(not(not(x)), y), implies(x, y))) cnf(prove_cn_{62}, negated_conjecture)

LCL397-1.p CN-63 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-63 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z)))) cnf(cn_1, axiom)
is_a_theorem(implies(implies(not(x), x), x)) cnf(cn_2, axiom)
is_a_theorem(implies(x, implies(not(x), y))) cnf(cn_3, axiom)
$\neg$ is_a_theorem(implies(x, implies(y, y))) cnf(prove_cn_{63}, negated_conjecture)

LCL398-1.p CN-64 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-64 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z)))) cnf(cn_1, axiom)
is_a_theorem(implies(implies(not(x), x), x)) cnf(cn_2, axiom)
is_a_theorem(implies(x, implies(not(x), y))) cnf(cn_3, axiom)
$\neg$ is_a_theorem(implies(not(implies(x, x)), y)) cnf(prove_cn_{64}, negated_conjecture)

LCL399-1.p CN-65 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-65 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z)))) cnf(cn_1, axiom)
is_a_theorem(implies(implies(not(x), x), x)) cnf(cn_2, axiom)
is_a_theorem(implies(x, implies(not(x), y))) cnf(cn_3, axiom)
$\neg$ is_a_theorem(implies(implies(not(x), not(implies(y, y))), x)) cnf(prove_cn_{65}, negated_conjecture)

LCL400-1.p CN-66 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-66 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z)))) cnf(cn_1, axiom)
is_a_theorem(implies(implies(not(x), x), x)) cnf(cn_2, axiom)
is_a_theorem(implies(x, implies(not(x), y))) cnf(cn_3, axiom)
$\neg$ is_a_theorem(implies(not(implies(x, y)), x)) cnf(prove_cn_{66}, negated_conjecture)

LCL401-1.p CN-67 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-67 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z)))) cnf(cn_1, axiom)
is_a_theorem(implies(implies(not(x), x), x)) cnf(cn_2, axiom)
is_a_theorem(implies(x, implies(not(x), y))) cnf(cn_3, axiom)
$\neg$ is_a_theorem(implies(not(implies(x, y)), not(y))) cnf(prove_cn_{67}, negated_conjecture)

LCL402-1.p CN-68 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-68 depends on the Lukasiewicz system.
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) $\Rightarrow$ is_a_theorem(y) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z)))) cnf(cn_1, axiom)
is_a_theorem(implies(implies(not(x), x), x)) cnf(cn_2, axiom)
is_a_theorem(implies(x, implies(not(x), y))) cnf(cn_3, axiom)
$\neg$ is_a_theorem(implies(not(implies(x, not(y))), y)) cnf(prove_cn_{68}, negated_conjecture)

LCL403-1.p CN-69 depends on the Lukasiewicz system

An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-69 depends on the Lukasiewicz system.
$(\text{is\_a\_theorem}(\text{implies}(x, y)) \text{ and } \text{is\_a\_theorem}(x)) \Rightarrow \text{is\_a\_theorem}(y)$ cnf(condensed_detachment, axiom)
$\text{is\_a\_theorem}(\text{implies}(\text{implies}(x, y), \text{implies}(\text{implies}(y, z), \text{implies}(x, z))))$ cnf(cn_1, axiom)
$\text{is\_a\_theorem}(\text{implies}(\text{implies}(\text{not}(x), x), x))$ cnf(cn_2, axiom)
$\text{is\_a\_theorem}(\text{implies}(x, \text{implies}(\text{not}(x), y)))$ cnf(cn_3, axiom)
$\neg\,\text{is\_a\_theorem}(\text{implies}(x, \text{implies}(\text{not}(y), \text{not}(\text{implies}(x, y)))))$ cnf(prove_cn_{69}, negated_conjecture)

LCL404-1.p CN-70 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-70 depends on the Lukasiewicz system.
$(\text{is\_a\_theorem}(\text{implies}(x, y)) \text{ and } \text{is\_a\_theorem}(x)) \Rightarrow \text{is\_a\_theorem}(y)$ cnf(condensed_detachment, axiom)
$\text{is\_a\_theorem}(\text{implies}(\text{implies}(x, y), \text{implies}(\text{implies}(y, z), \text{implies}(x, z))))$ cnf(cn_1, axiom)
$\text{is\_a\_theorem}(\text{implies}(\text{implies}(\text{not}(x), x), x))$ cnf(cn_2, axiom)
$\text{is\_a\_theorem}(\text{implies}(x, \text{implies}(\text{not}(x), y)))$ cnf(cn_3, axiom)
$\neg\,\text{is\_a\_theorem}(\text{implies}(x, \text{implies}(y, \text{not}(\text{implies}(x, \text{not}(y))))))$ cnf(prove_cn_{70}, negated_conjecture)

LCL405-1.p CN-71 depends on the Lukasiewicz system
An axiomatisation of the Implication/Negation 2 valued sentential calculus is CN-1,CN-2,CN-3 by Lukasiewicz. Show that CN-71 depends on the Lukasiewicz system.
$(\text{is\_a\_theorem}(\text{implies}(x, y)) \text{ and } \text{is\_a\_theorem}(x)) \Rightarrow \text{is\_a\_theorem}(y)$ cnf(condensed_detachment, axiom)
$\text{is\_a\_theorem}(\text{implies}(\text{implies}(x, y), \text{implies}(\text{implies}(y, z), \text{implies}(x, z))))$ cnf(cn_1, axiom)
$\text{is\_a\_theorem}(\text{implies}(\text{implies}(\text{not}(x), x), x))$ cnf(cn_2, axiom)
$\text{is\_a\_theorem}(\text{implies}(x, \text{implies}(\text{not}(x), y)))$ cnf(cn_3, axiom)
$\neg\,\text{is\_a\_theorem}(\text{not}(\text{implies}(\text{implies}(x, x), \text{not}(\text{implies}(y, y)))))$ cnf(prove_cn_{71}, negated_conjecture)

LCL406-1.p Generate LTL structures of size 4
$x \leq x$ cnf(reflexivity_of_less_or_equal, axiom)
$(x \leq y \text{ and } y \leq z) \Rightarrow x \leq z$ cnf(transitivity_of_less_or_equal, axiom)
$x \leq y \text{ or } y \leq x$ cnf(completeness_of_less_or_equal, axiom)
$\text{successor}(x) = y \Rightarrow x \leq y$ cnf(predecessor_less_or_equal, axiom)
$x \leq y \Rightarrow (y = x \text{ or } y = \text{successor}(x) \text{ or } y = \text{successor}(\text{successor}(x)) \text{ or } y = \text{successor}(\text{successor}(\text{successor}(x))))$ cnf(four_s

LCL407-1.p Wajsberg algebra axioms
include('Axioms/LCL001-0.ax')

LCL407-2.p Alternative Wajsberg algebra axioms
include('Axioms/LCL002-0.ax')

LCL408-1.p Wajsberg algebra lattice structure definitions
include('Axioms/LCL001-0.ax')
include('Axioms/LCL001-1.ax')

LCL409-1.p Wajsberg algebra AND and OR definitions
include('Axioms/LCL001-0.ax')
include('Axioms/LCL001-2.ax')

LCL410-1.p Alternative Wajsberg algebra definitions
include('Axioms/LCL002-0.ax')
include('Axioms/LCL001-2.ax')
include('Axioms/LCL002-1.ax')

LCL411-1.p Propositional logic deduction axioms
include('Axioms/LCL003-0.ax')

LCL411-2.p Propositional logic deduction axioms
include('Axioms/LCL004-0.ax')

LCL412-1.p Propositional logic deduction axioms for AND
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')

LCL413-1.p Propositional logic deduction axioms for EQUIVALENT
include('Axioms/LCL004-0.ax')
include('Axioms/LCL004-1.ax')
include('Axioms/LCL004-2.ax')

LCL414+1.p Peter Andrews Problem THM147
$\neg \forall p, q$: ($\neg$ a_truth(implies(p, q)) or $\neg$ a_truth(p) or a_truth(q)) and $\forall p, q$: a_truth(implies(p, implies(q, p))) and $\forall p, q, r$: a_trutl

LCL414-1.p Peter Andrews Problem THM147
(a_truth(a) and a_truth(implies(a, b))) $\Rightarrow$ a_truth(b) cnf(thm147$_1$, negated_conjecture)
a_truth(implies(a, implies(b, a))) cnf(thm147$_2$, negated_conjecture)
a_truth(implies(implies(a, implies(b, c)), implies(implies(a, b), implies(a, c)))) cnf(thm147$_3$, negated_conjecture)
a_truth(implies(implies(not(a), not(b)), implies(b, a))) cnf(thm147$_4$, negated_conjecture)
$\neg$ a_truth(implies(sk$_1$, sk$_1$)) cnf(thm147$_5$, negated_conjecture)

LCL414^7.p Peter Andrews Problem THM147
include('Axioms/LCL015^0.ax')
include('Axioms/LCL013^5.ax')
include('Axioms/LCL015^1.ax')
a_truth: mu $\rightarrow$ \$i $\rightarrow$ \$o thf(a_truth_type, type)
not: mu $\rightarrow$ mu thf(not_type, type)
$\forall v$: \$i, v_1: mu: (exists_in_world@(not@v_1)@v) thf(existence_of_not_ax, axiom)
implies: mu $\rightarrow$ mu $\rightarrow$ mu thf(implies_type, type)
$\forall v$: \$i, v_2: mu, v_1: mu: (exists_in_world@(implies@v_2@v_1)@v) thf(existence_of_implies_ax, axiom)
mvalid@(mbox_s$_4$@(mnot@(mand@(mbox_s$_4$@(mforall_ind@λp: mu: (mbox_s$_4$@(mforall_ind@λq: mu: (mor@(mbox_s$_4$@(mnot

LCL415-1.p Non-axiom for intuitionistic implication
Show that the candidate formula is not a single axiom for intuitionistic implication by finding a model in which the required property fails.
(is_a_theorem(implies(a, b)) and is_a_theorem(a)) $\Rightarrow$ is_a_theorem(b) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(a, b), implies(implies(b, implies(implies(c, a), d)), implies(a, d)))) cnf(candidate, axiom)
$\neg$ is_a_theorem(implies(a, implies(b, a))) cnf(prove_required_property, negated_conjecture)

LCL416-1.p Prove reflexivity from formula XCB by condensed detachment
(is_a_theorem(equivalent(a, b)) and is_a_theorem(a)) $\Rightarrow$ is_a_theorem(b) cnf(condensed_detachment, axiom)
is_a_theorem(equivalent(a, equivalent(equivalent(equivalent(a, b), equivalent(c, b)), c))) cnf(xcb, axiom)
$\neg$ is_a_theorem(equivalent(a, a)) cnf(prove_reflexivity, negated_conjecture)

LCL417-1.p XCB is a single axiom for the equivalential calculus
Show that formula XCB is a single axiom for the equivalential calculus by deriving the single axiom WN by condensed detachment.
(is_a_theorem(equivalent(a, b)) and is_a_theorem(a)) $\Rightarrow$ is_a_theorem(b) cnf(condensed_detachment, axiom)
is_a_theorem(equivalent(a, equivalent(equivalent(equivalent(a, b), equivalent(c, b)), c))) cnf(xcb, axiom)
$\neg$ is_a_theorem(equivalent(equivalent(a, equivalent(b, c)), equivalent(c, equivalent(a, b)))) cnf(prove_wn, negated_conjecture

LCL417-2.p XCB is a single axiom for the equivalential calculus
Show that formula XCB is a single axiom for the equivalential calculus by deriving the pair of axioms symmetry, transitivity.
(is_a_theorem(equivalent(a, b)) and is_a_theorem(a)) $\Rightarrow$ is_a_theorem(b) cnf(condensed_detachment, axiom)
is_a_theorem(equivalent(a, equivalent(equivalent(equivalent(a, b), equivalent(c, b)), c))) cnf(xcb, axiom)
is_a_theorem(equivalent(equivalent(a, b), equivalent(b, a))) $\Rightarrow$ $\neg$ is_a_theorem(equivalent(equivalent(a, b), equivalent(equivalen

LCL418-1.p Is formula YQE a single axiom for the right group calculus?
(is_a_theorem(equivalent(a, b)) and is_a_theorem(a)) $\Rightarrow$ is_a_theorem(b) cnf(condensed_detachment, axiom)
is_a_theorem(equivalent(equivalent(a, b), equivalent(equivalent(a, c), equivalent(b, c)))) cnf(yqe, axiom)
$\neg$ is_a_theorem(equivalent(a, equivalent(a, equivalent(equivalent(b, c), equivalent(equivalent(b, d), equivalent(c, d)))))) cnf(p

LCL419-1.p Prove AK1 from MV1–MV4
(is_a_theorem(implies(a, b)) and is_a_theorem(a)) $\Rightarrow$ is_a_theorem(b) cnf(condensed_detachment, axiom)
is_a_theorem(implies(a, implies(b, a))) cnf(mv$_1$, axiom)
is_a_theorem(implies(implies(a, b), implies(implies(b, c), implies(a, c)))) cnf(mv$_2$, axiom)
is_a_theorem(implies(implies(implies(a, b), b), implies(implies(b, a), a))) cnf(mv$_3$, axiom)
is_a_theorem(implies(implies(not(a), not(b)), implies(b, a))) cnf(mv$_4$, axiom)
$\neg$ is_a_theorem(implies(implies(implies(a, not(implies(implies(not(b), not(c)), not(c)))), not(implies(implies(not(b), not(c)), not

LCL420-1.p Prove AK2 from MV1–MV4
(is_a_theorem(implies(a, b)) and is_a_theorem(a)) $\Rightarrow$ is_a_theorem(b) cnf(condensed_detachment, axiom)
is_a_theorem(implies(a, implies(b, a))) cnf(mv$_1$, axiom)
is_a_theorem(implies(implies(a, b), implies(implies(b, c), implies(a, c)))) cnf(mv$_2$, axiom)
is_a_theorem(implies(implies(implies(a, b), b), implies(implies(b, a), a))) cnf(mv$_3$, axiom)

is_a_theorem(implies(implies(not(a), not(b)), implies(b, a))) cnf(mv$_4$, axiom)
¬ is_a_theorem(implies(not(implies(implies(not(implies(implies(a, b), b)), not(implies(implies(a, c), c))), not(implies(implies(a,

LCL421-1.p Prove KA1 from MV1–MV4
(is_a_theorem(implies(a, b)) and is_a_theorem(a)) ⇒ is_a_theorem(b) cnf(condensed_detachment, axiom)
is_a_theorem(implies(a, implies(b, a))) cnf(mv$_1$, axiom)
is_a_theorem(implies(implies(a, b), implies(implies(b, c), implies(a, c)))) cnf(mv$_2$, axiom)
is_a_theorem(implies(implies(implies(a, b), b), implies(implies(b, a), a))) cnf(mv$_3$, axiom)
is_a_theorem(implies(implies(not(a), not(b)), implies(b, a))) cnf(mv$_4$, axiom)
¬ is_a_theorem(implies(not(implies(implies(not(a), not(implies(implies(b, c), c))), not(implies(implies(b, c), c)))), implies(implie

LCL422-1.p Prove KA2 from MV1–MV4
(is_a_theorem(implies(a, b)) and is_a_theorem(a)) ⇒ is_a_theorem(b) cnf(condensed_detachment, axiom)
is_a_theorem(implies(a, implies(b, a))) cnf(mv$_1$, axiom)
is_a_theorem(implies(implies(a, b), implies(implies(b, c), implies(a, c)))) cnf(mv$_2$, axiom)
is_a_theorem(implies(implies(implies(a, b), b), implies(implies(b, a), a))) cnf(mv$_3$, axiom)
is_a_theorem(implies(implies(not(a), not(b)), implies(b, a))) cnf(mv$_4$, axiom)
¬ is_a_theorem(implies(implies(implies(not(implies(implies(not(a), not(b)), not(b))), not(implies(implies(not(a), not(c)), not(c)

LCL423-1.p Luka-23 is a single axiom
Show that the formula Luka-23 is a single axiom for two-valued logic by deriving the Lukasiewicz 3-basis.
(is_a_theorem(implies(a, b)) and is_a_theorem(a)) ⇒ is_a_theorem(b) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(implies(a, b), implies(implies(implies(not(c), not(d)), e), c)), implies(f, implies(implies(c, a), impli
(is_a_theorem(implies(implies(a, b), implies(implies(b, c), implies(a, c)))) and is_a_theorem(implies(implies(not(a), a), a))) ⇒
¬ is_a_theorem(implies(a, implies(not(a), b))) cnf(prove_luka_3_basis, negated_conjecture)

LCL424-1.p Mer-21 is a single axiom for two-valued logic
Show that the formula Mer-21 is a single axiom for two-valued logic by deriving the Lukasiewicz 3-basis.
(is_a_theorem(implies(a, b)) and is_a_theorem(a)) ⇒ is_a_theorem(b) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(implies(implies(implies(a, b), implies(not(c), not(d))), c), e), implies(implies(e, a), implies(d, a))))
(is_a_theorem(implies(implies(a, b), implies(implies(b, c), implies(a, c)))) and is_a_theorem(implies(implies(not(a), a), a))) ⇒
¬ is_a_theorem(implies(a, implies(not(a), b))) cnf(prove_luka_3_basis, negated_conjecture)

LCL425-1.p BCI+mingle implies Karpenko by condensed detachment
Show that if the mingle formula is added to the logic BCI, the Karpenko formula can be derived by condensed detachment.
(is_a_theorem(implies(a, b)) and is_a_theorem(a)) ⇒ is_a_theorem(b) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(a, b), implies(implies(c, a), implies(c, b)))) cnf(b, axiom)
is_a_theorem(implies(implies(a, implies(b, c)), implies(b, implies(a, c)))) cnf(c, axiom)
is_a_theorem(implies(implies(implies(implies(implies(a, b), b), a), c), implies(implies(implies(implies(implies(b, a), a), b), c), c)))
¬ is_a_theorem(implies(implies(a, implies(implies(b, b), a)), implies(implies(implies(a, b), b), implies(implies(b, a), a)))) cnf(

LCL426-1.p Prove the mingle formula by condensed detachment
Show that the mingle axiom can be derived from the three formulas given below by condensed detachment.
(is_a_theorem(implies(a, b)) and is_a_theorem(a)) ⇒ is_a_theorem(b) cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(a, b), implies(implies(b, c), implies(a, c)))) cnf(suffixing, axiom)
is_a_theorem(implies(a, implies(implies(a, b), b))) cnf(assertion, axiom)
is_a_theorem(implies(implies(implies(implies(implies(a, b), c), implies(b, a)), c), c)) cnf(candidate, axiom)
¬ is_a_theorem(implies(implies(implies(implies(implies(a, b), b), a), c), implies(implies(implies(implies(implies(b, a), a), b), c), c)

LCL427-1.p ORG-D23 is a single axiom for propositional calculus
Show that formula ORG-D23 is a single axiom for propositional calculus in terms of the Sheffer stroke by deriving Nicod's single axiom.
(is_a_theorem(detach(a, detach(b, c))) and is_a_theorem(a)) ⇒ is_a_theorem(c) cnf(condensed_detachment, axiom)
is_a_theorem(detach(detach(a, detach(b, c)), detach(detach(a, detach(b, c)), detach(detach(d, c), detach(detach(c, d), detach(a,
¬ is_a_theorem(detach(detach(a, detach(b, c)), detach(detach(e, detach(e, e)), detach(detach(f, b), detach(detach(a, f), detach(

LCL428-1.p Prove the Harris/Rezus axiom from MV1-MV3 and MV5
(is_a_theorem(implies(a, b)) and is_a_theorem(a)) ⇒ is_a_theorem(b) cnf(condensed_detachment, axiom)
is_a_theorem(implies(a, implies(b, a))) cnf(mv$_1$, axiom)
is_a_theorem(implies(implies(a, b), implies(implies(b, c), implies(a, c)))) cnf(mv$_2$, axiom)
is_a_theorem(implies(implies(implies(a, b), b), implies(implies(b, a), a))) cnf(mv$_3$, axiom)
is_a_theorem(implies(implies(implies(x, y), implies(y, x)), implies(y, x))) cnf(mv$_5$, axiom)
¬ is_a_theorem(implies(implies(implies(x, implies(y, x)), implies(implies(implies(implies(implies(implies(implies(implie

LCL429-2.p Problem about propositional logic
(c_in(v_p, c_PropLog_Othms(c_minus(v_A, v_B, tc_set(tc_PropLog_Opl(t_a))), t_a), tc_PropLog_Opl(t_a)) and c_lessequals(v_A
c_in(v_p, c_PropLog_Othms(c_minus(v_C, v_B, tc_set(tc_PropLog_Opl(t_a))), t_a), tc_PropLog_Opl(t_a))   cnf(cls_PropLog_O
c_lessequals(c_PropLog_Ohyps(v_p, c_minus(v_t, c_insert(v_v, c_emptyset, t_a), tc_set(t_a)), t_a), c_insert(c_PropLog_Opl_Oop_A
c_lessequals(c_minus(v_B, v_C, tc_set(t_a)), c_insert(v_a, c_minus(v_B, c_insert(v_a, v_C, t_a), tc_set(t_a)), t_a), tc_set(t_a))   cn
c_lessequals(c_minus(c_insert(v_a, c_minus(v_B, c_insert(v_c, c_emptyset, t_a), tc_set(t_a)), t_a), v_D, tc_set(t_a)), c_insert(v_a, c_
(c_in(v_q, c_PropLog_Othms(c_insert(v_p, v_H, tc_PropLog_Opl(t_a)), t_a), tc_PropLog_Opl(t_a)) and c_in(v_q, c_PropLog_Othm
c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a))   cnf(cls_PropLog_Othms__excluded__middle__rule$_0$, axiom)
(c_in(v_p, c_PropLog_Othms(v_G, t_a), tc_PropLog_Opl(t_a)) and c_lessequals(v_G, v_H, tc_set(tc_PropLog_Opl(t_a)))) ⇒
c_in(v_p, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a))   cnf(cls_PropLog_Oweaken__left$_0$, axiom)
c_in(v_p, c_PropLog_Othms(c_minus(c_PropLog_Ohyps(v_p, v_U, t_a), v_F, tc_set(tc_PropLog_Opl(t_a))), t_a), tc_PropLog_Opl(
¬ c_in(v_p, c_PropLog_Othms(c_minus(c_PropLog_Ohyps(v_p, v_t, t_a), c_insert(c_PropLog_Opl_Ovar(v_v, t_a), v_F, tc_PropLog

LCL430-2.p Problem about propositional logic
(c_in(v_p, c_PropLog_Othms(c_minus(v_A, v_B, tc_set(tc_PropLog_Opl(t_a))), t_a), tc_PropLog_Opl(t_a)) and c_lessequals(v_A
c_in(v_p, c_PropLog_Othms(c_minus(v_C, v_B, tc_set(tc_PropLog_Opl(t_a))), t_a), tc_PropLog_Opl(t_a))   cnf(cls_PropLog_O
c_lessequals(c_PropLog_Ohyps(v_p, c_insert(v_v, v_t, t_a), t_a), c_insert(c_PropLog_Opl_Ovar(v_v, t_a), c_minus(c_PropLog_Ohy
c_lessequals(c_minus(v_B, v_C, tc_set(t_a)), c_insert(v_a, c_minus(v_B, c_insert(v_a, v_C, t_a), tc_set(t_a)), t_a), tc_set(t_a))   cn
c_lessequals(c_minus(c_insert(v_a, c_minus(v_B, c_insert(v_c, c_emptyset, t_a), tc_set(t_a)), t_a), v_D, tc_set(t_a)), c_insert(v_a, c_
(c_in(v_q, c_PropLog_Othms(c_insert(v_p, v_H, tc_PropLog_Opl(t_a)), t_a), tc_PropLog_Opl(t_a)) and c_in(v_q, c_PropLog_Othm
c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a))   cnf(cls_PropLog_Othms__excluded__middle__rule$_0$, axiom)
(c_in(v_p, c_PropLog_Othms(v_G, t_a), tc_PropLog_Opl(t_a)) and c_lessequals(v_G, v_H, tc_set(tc_PropLog_Opl(t_a)))) ⇒
c_in(v_p, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a))   cnf(cls_PropLog_Oweaken__left$_0$, axiom)
c_in(v_p, c_PropLog_Othms(c_minus(c_PropLog_Ohyps(v_p, v_U, t_a), v_F, tc_set(tc_PropLog_Opl(t_a))), t_a), tc_PropLog_Opl(
¬ c_in(v_p, c_PropLog_Othms(c_minus(c_PropLog_Ohyps(v_p, v_t, t_a), c_insert(c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Ov

LCL431-2.p Problem about propositional logic
c_PropLog_Osat(c_emptyset, v_p, t_a)   cnf(cls_conjecture$_0$, negated_conjecture)
¬ c_in(v_p, c_PropLog_Othms(c_emptyset, t_a), tc_PropLog_Opl(t_a))   cnf(cls_conjecture$_1$, negated_conjecture)
c_PropLog_Osat(c_emptyset, v_p, t_a) ⇒ c_in(v_p, c_PropLog_Othms(c_minus(c_PropLog_Ohyps(v_p, v_U, t_a), c_PropLog_Oh
c_minus(v_A, v_A, tc_set(t_a)) = c_emptyset   cnf(cls_Set_ODiff__cancel$_0$, axiom)

LCL432-1.p Problem about propositional logic
include('Axioms/LCL005-0.ax')
include('Axioms/MSC001-2.ax')
include('Axioms/MSC001-0.ax')
c_PropLog_Osat(c_emptyset, v_p, t_a) ⇒ c_in(v_p, c_PropLog_Othms(c_emptyset, t_a), tc_PropLog_Opl(t_a))   cnf(cls_PropL
c_in(v_q, c_PropLog_Othms(c_insert(v_p, v_H, tc_PropLog_Opl(t_a)), t_a), tc_PropLog_Opl(t_a)) ⇒ c_in(c_PropLog_Opl_Oop_
c_PropLog_Osat(c_insert(v_p, v_H, tc_PropLog_Opl(t_a)), v_q, t_a) ⇒ c_PropLog_Osat(v_H, c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v
(c_in(v_p, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) and c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), c_PropLog_Ot
c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a))   cnf(cls_PropLog_Othms_OMP$_0$, axiom)
(c_in(v_p, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) and c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), v_H, tc_PropL
c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a))   cnf(cls_PropLog_Othms__H__MP$_0$, axiom)
c_in(v_p, c_PropLog_Othms(v_G, t_a), tc_PropLog_Opl(t_a)) ⇒ c_in(v_p, c_PropLog_Othms(c_insert(v_a, v_G, tc_PropLog_Opl
c_PropLog_Osat(c_emptyset, v_x, t_a)   cnf(cls_conjecture$_0$, negated_conjecture)
¬ c_in(v_x, c_PropLog_Othms(c_emptyset, t_a), tc_PropLog_Opl(t_a))   cnf(cls_conjecture$_1$, negated_conjecture)

LCL432-2.p Problem about propositional logic
c_PropLog_Osat(c_emptyset, v_x, t_a)   cnf(cls_conjecture$_0$, negated_conjecture)
¬ c_in(v_x, c_PropLog_Othms(c_emptyset, t_a), tc_PropLog_Opl(t_a))   cnf(cls_conjecture$_1$, negated_conjecture)
c_PropLog_Osat(c_emptyset, v_p, t_a) ⇒ c_in(v_p, c_PropLog_Othms(c_emptyset, t_a), tc_PropLog_Opl(t_a))   cnf(cls_PropL

LCL433-1.p Problem about propositional logic
include('Axioms/LCL005-0.ax')
include('Axioms/MSC001-2.ax')
include('Axioms/MSC001-0.ax')
c_PropLog_Osat(c_emptyset, v_p, t_a) ⇒ c_in(v_p, c_PropLog_Othms(c_emptyset, t_a), tc_PropLog_Opl(t_a))   cnf(cls_PropL
c_in(v_q, c_PropLog_Othms(c_insert(v_p, v_H, tc_PropLog_Opl(t_a)), t_a), tc_PropLog_Opl(t_a)) ⇒ c_in(c_PropLog_Opl_Oop_
c_PropLog_Osat(c_insert(v_p, v_H, tc_PropLog_Opl(t_a)), v_q, t_a) ⇒ c_PropLog_Osat(v_H, c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v
(c_in(v_p, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) and c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), c_PropLog_Ot
c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a))   cnf(cls_PropLog_Othms_OMP$_0$, axiom)

(c_in(v_p, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) and c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), v_H, tc_PropL
c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_PropLog_Othms__H__MP$_0$, axiom)
c_in(v_p, c_PropLog_Othms(v_G, t_a), tc_PropLog_Opl(t_a)) ⇒ c_in(v_p, c_PropLog_Othms(c_insert(v_a, v_G, tc_PropLog_Opl
c_in(v_F, c_Finite__Set_OFinites, tc_set(tc_PropLog_Opl(t_a))) cnf(cls_conjecture$_0$, negated_conjecture)
¬ c_in(v_x, v_F, tc_PropLog_Opl(t_a)) cnf(cls_conjecture$_1$, negated_conjecture)
c_PropLog_Osat(c_insert(v_x, v_F, tc_PropLog_Opl(t_a)), v_xa, t_a) cnf(cls_conjecture$_2$, negated_conjecture)
¬ c_in(v_xa, c_PropLog_Othms(c_insert(v_x, v_F, tc_PropLog_Opl(t_a)), t_a), tc_PropLog_Opl(t_a)) cnf(cls_conjecture$_3$, neg
c_PropLog_Osat(v_F, v_U, t_a) ⇒ c_in(v_U, c_PropLog_Othms(v_F, t_a), tc_PropLog_Opl(t_a)) cnf(cls_conjecture$_4$, negate

LCL433-2.p Problem about propositional logic
c_PropLog_Osat(c_insert(v_x, v_F, tc_PropLog_Opl(t_a)), v_xa, t_a) cnf(cls_conjecture$_2$, negated_conjecture)
¬ c_in(v_xa, c_PropLog_Othms(c_insert(v_x, v_F, tc_PropLog_Opl(t_a)), t_a), tc_PropLog_Opl(t_a)) cnf(cls_conjecture$_3$, neg
c_PropLog_Osat(v_F, v_U, t_a) ⇒ c_in(v_U, c_PropLog_Othms(v_F, t_a), tc_PropLog_Opl(t_a)) cnf(cls_conjecture$_4$, negate
c_PropLog_Osat(c_insert(v_p, v_H, tc_PropLog_Opl(t_a)), v_q, t_a) ⇒ c_PropLog_Osat(v_H, c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v
c_in(v_p, v_H, tc_PropLog_Opl(t_a)) ⇒ c_in(v_p, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_PropLog_Othn
(c_in(v_p, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) and c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), c_PropLog_Ot
c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_PropLog_Othms_OMP$_0$, axiom)
c_in(v_p, c_PropLog_Othms(v_G, t_a), tc_PropLog_Opl(t_a)) ⇒ c_in(v_p, c_PropLog_Othms(c_insert(v_a, v_G, tc_PropLog_Opl
c_in(v_x, c_insert(v_x, v_B, t_a), t_a) cnf(cls_Set_OinsertCI$_1$, axiom)

LCL434-1.p Problem about propositional logic
include('Axioms/LCL005-0.ax')
include('Axioms/MSC001-2.ax')
include('Axioms/MSC001-0.ax')
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, c_PropLog_Opl_Ofalse, t_a), c_Pr
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, c_PropLog_Opl_Oop_A_N$_{62}$(v_q, v_p, t_a), t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_C
(c_in(v_p, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) and c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), c_PropLog_Ot
c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_PropLog_Othms_OMP$_0$, axiom)
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, c_PropLog_Opl_Oop_A_N$_{62}$(v_q, v_r, t_a), t_a), c_PropLog_Op
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_p, t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_PropLog_Othms_
c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) ⇒ c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), c_PropLog_Oth
c_in(v_pa, c_insert(v_p, v_H, tc_PropLog_Opl(t_a)), tc_PropLog_Opl(t_a)) cnf(cls_conjecture$_0$, negated_conjecture)
¬ c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_pa, t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_conjecture$_1$, n

LCL434-2.p Problem about propositional logic
c_in(v_pa, c_insert(v_p, v_H, tc_PropLog_Opl(t_a)), tc_PropLog_Opl(t_a)) cnf(cls_conjecture$_0$, negated_conjecture)
¬ c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_pa, t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_conjecture$_1$, n
c_in(v_p, v_H, tc_PropLog_Opl(t_a)) ⇒ c_in(v_p, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_PropLog_Othn
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_p, t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_PropLog_Othms_
c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) ⇒ c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), c_PropLog_Oth
c_in(v_a, c_insert(v_b, v_A, t_a), t_a) ⇒ (c_in(v_a, v_A, t_a) or v_a = v_b) cnf(cls_Set_OinsertE$_0$, axiom)

LCL435-1.p Problem about propositional logic
include('Axioms/LCL005-0.ax')
include('Axioms/MSC001-2.ax')
include('Axioms/MSC001-0.ax')
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, c_PropLog_Opl_Ofalse, t_a), c_Pr
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, c_PropLog_Opl_Oop_A_N$_{62}$(v_q, v_p, t_a), t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_C
(c_in(v_p, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) and c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), c_PropLog_Ot
c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_PropLog_Othms_OMP$_0$, axiom)
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, c_PropLog_Opl_Oop_A_N$_{62}$(v_q, v_r, t_a), t_a), c_PropLog_Op
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_p, t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_PropLog_Othms_
c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) ⇒ c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), c_PropLog_Oth
¬ c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, c_PropLog_Opl_Oop_A_N$_{62}$(v_pa, c_PropLog_Opl_Oop_A_N$_{62}$(v_q, v_pa, t_a), t_a), t_a),

LCL435-2.p Problem about propositional logic
¬ c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, c_PropLog_Opl_Oop_A_N$_{62}$(v_pa, c_PropLog_Opl_Oop_A_N$_{62}$(v_q, v_pa, t_a), t_a), t_a),
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, c_PropLog_Opl_Oop_A_N$_{62}$(v_q, v_p, t_a), t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_C
c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) ⇒ c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), c_PropLog_Oth

LCL436-1.p Problem about propositional logic
include('Axioms/LCL005-0.ax')
include('Axioms/MSC001-2.ax')

include('Axioms/MSC001-0.ax')
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, c_PropLog_Opl_Ofalse, t_a), c_Pr
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, c_PropLog_Opl_Oop_A_N$_{62}$(v_q, v_p, t_a), t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_O
(c_in(v_p, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) and c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), c_PropLog_Ot
c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a))   cnf(cls_PropLog_Othms_OMP$_0$, axiom)
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, c_PropLog_Opl_Oop_A_N$_{62}$(v_q, v_r, t_a), t_a), c_PropLog_Op
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_p, t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a))   cnf(cls_PropLog_Othms_
c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) $\Rightarrow$ c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), c_PropLog_Oth
¬ c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_A_N$_{62}$(v_pa, c_PropLog_Opl_Oop_A_N

LCL436-2.p Problem about propositional logic
¬ c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_A_N$_{62}$(v_pa, c_PropLog_Opl_Oop_A_N
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, c_PropLog_Opl_Oop_A_N$_{62}$(v_q, v_r, t_a), t_a), c_PropLog_Op
c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) $\Rightarrow$ c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), c_PropLog_Oth

LCL437-1.p Problem about propositional logic
include('Axioms/LCL005-0.ax')
include('Axioms/MSC001-2.ax')
include('Axioms/MSC001-0.ax')
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, c_PropLog_Opl_Ofalse, t_a), c_Pr
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, c_PropLog_Opl_Oop_A_N$_{62}$(v_q, v_p, t_a), t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_O
(c_in(v_p, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) and c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), c_PropLog_Ot
c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a))   cnf(cls_PropLog_Othms_OMP$_0$, axiom)
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, c_PropLog_Opl_Oop_A_N$_{62}$(v_q, v_r, t_a), t_a), c_PropLog_Op
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_p, t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a))   cnf(cls_PropLog_Othms_
c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) $\Rightarrow$ c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), c_PropLog_Oth
¬ c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_A_N$_{62}$(v_

LCL437-2.p Problem about propositional logic
¬ c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_A_N$_{62}$(v_
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, c_PropLog_Opl_Ofalse, t_a), c_Pr
c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) $\Rightarrow$ c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), c_PropLog_Oth

LCL438-1.p Problem about propositional logic
include('Axioms/LCL005-0.ax')
include('Axioms/MSC001-2.ax')
include('Axioms/MSC001-0.ax')
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, c_PropLog_Opl_Ofalse, t_a), c_Pr
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, c_PropLog_Opl_Oop_A_N$_{62}$(v_q, v_p, t_a), t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_O
(c_in(v_p, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) and c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), c_PropLog_Ot
c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a))   cnf(cls_PropLog_Othms_OMP$_0$, axiom)
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, c_PropLog_Opl_Oop_A_N$_{62}$(v_q, v_r, t_a), t_a), c_PropLog_Op
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_p, t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a))   cnf(cls_PropLog_Othms_
c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) $\Rightarrow$ c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), c_PropLog_Oth
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_pa, v_q, t_a), c_PropLog_Othms(c_insert(v_p, v_H, tc_PropLog_Opl(t_a)), t_a), tc_PropLog_O
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, c_PropLog_Opl_Oop_A_N$_{62}$(v_pa, v_q, t_a), t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_
c_in(v_pa, c_PropLog_Othms(c_insert(v_p, v_H, tc_PropLog_Opl(t_a)), t_a), tc_PropLog_Opl(t_a))   cnf(cls_conjecture$_2$, negat
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_pa, t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a))   cnf(cls_conjecture$_3$, neg
¬ c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a))   cnf(cls_conjecture$_4$, ne

LCL438-2.p Problem about propositional logic
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, c_PropLog_Opl_Oop_A_N$_{62}$(v_pa, v_q, t_a), t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_pa, t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a))   cnf(cls_conjecture$_3$, neg
¬ c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a))   cnf(cls_conjecture$_4$, ne
(c_in(v_p, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) and c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), c_PropLog_Ot
c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a))   cnf(cls_PropLog_Othms_OMP$_0$, axiom)
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, c_PropLog_Opl_Oop_A_N$_{62}$(v_q, v_r, t_a), t_a), c_PropLog_Op

LCL439-1.p Problem about propositional logic
include('Axioms/LCL005-0.ax')
include('Axioms/MSC001-2.ax')
include('Axioms/MSC001-0.ax')
c_in(v_q, c_PropLog_Othms(c_insert(v_p, v_H, tc_PropLog_Opl(t_a)), t_a), tc_PropLog_Opl(t_a)) $\Rightarrow$ c_in(c_PropLog_Opl_Oop_

(c_in(v_p, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) and c_in(c_PropLog_Opl_Oop_A_N_{62}(v_p, c_PropLog_Opl_Ofalse,
c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_PropLog_Othms__notE$_0$, axiom)
c_in(v_p, c_PropLog_Othms(v_G, t_a), tc_PropLog_Opl(t_a)) ⇒ c_in(v_p, c_PropLog_Othms(c_insert(v_a, v_G, tc_PropLog_Opl
c_in(c_PropLog_Opl_Oop_A_N_{62}(v_p, c_PropLog_Opl_Ofalse, t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cl
¬ c_in(c_PropLog_Opl_Oop_A_N_{62}(v_p, v_q, t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_conjecture$_1$, neg

LCL439-2.p Problem about propositional logic
c_in(v_q, c_PropLog_Othms(c_insert(v_p, v_H, tc_PropLog_Opl(t_a)), t_a), tc_PropLog_Opl(t_a)) ⇒ c_in(c_PropLog_Opl_Oop_
c_in(v_p, v_H, tc_PropLog_Opl(t_a)) ⇒ c_in(v_p, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_PropLog_Oth
(c_in(v_p, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) and c_in(c_PropLog_Opl_Oop_A_N_{62}(v_p, c_PropLog_Opl_Ofalse,
c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_PropLog_Othms__notE$_0$, axiom)
c_in(v_p, c_PropLog_Othms(v_G, t_a), tc_PropLog_Opl(t_a)) ⇒ c_in(v_p, c_PropLog_Othms(c_insert(v_a, v_G, tc_PropLog_Opl
c_in(v_x, c_insert(v_x, v_B, t_a), t_a) cnf(cls_Set_OinsertCI$_1$, axiom)
c_in(c_PropLog_Opl_Oop_A_N_{62}(v_p, c_PropLog_Opl_Ofalse, t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cl
¬ c_in(c_PropLog_Opl_Oop_A_N_{62}(v_p, v_q, t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_conjecture$_1$, neg

LCL440-1.p Problem about propositional logic
include('Axioms/LCL005-0.ax')
include('Axioms/MSC001-2.ax')
include('Axioms/MSC001-0.ax')
c_in(c_PropLog_Opl_Oop_A_N_{62}(v_p, c_PropLog_Opl_Ofalse, t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) ⇒
c_in(c_PropLog_Opl_Oop_A_N_{62}(v_p, v_q, t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_PropLog_Ofalse_
(c_in(v_p, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) and c_in(c_PropLog_Opl_Oop_A_N_{62}(v_q, c_PropLog_Opl_Ofalse,
c_in(c_PropLog_Opl_Oop_A_N_{62}(c_PropLog_Opl_Oop_A_N_{62}(v_p, v_q, t_a), c_PropLog_Opl_Ofalse, t_a), c_PropLog_Othms(v_H
c_in(c_PropLog_Opl_Oop_A_N_{62}(v_p, v_p, t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_PropLog_Othms_
c_in(v_p, c_PropLog_Othms(v_G, t_a), tc_PropLog_Opl(t_a)) ⇒ c_in(v_p, c_PropLog_Othms(c_union(v_G, v_B, tc_PropLog_Op
c_in(v_p, c_PropLog_Othms(v_G, t_a), tc_PropLog_Opl(t_a)) ⇒ c_in(v_p, c_PropLog_Othms(c_union(v_A, v_G, tc_PropLog_Op
c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) ⇒ c_in(c_PropLog_Opl_Oop_A_N_{62}(v_p, v_q, t_a), c_PropLog_Oth
¬ c_in(c_PropLog_Opl_Oop_A_N_{62}(c_PropLog_Opl_Ofalse, c_PropLog_Opl_Ofalse, t_a), c_PropLog_Othms(c_emptyset, t_a), tc_F

LCL440-2.p Problem about propositional logic
¬ c_in(c_PropLog_Opl_Oop_A_N_{62}(c_PropLog_Opl_Ofalse, c_PropLog_Opl_Ofalse, t_a), c_PropLog_Othms(c_emptyset, t_a), tc_F
c_in(c_PropLog_Opl_Oop_A_N_{62}(v_p, v_p, t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_PropLog_Othms_

LCL441-1.p Problem about propositional logic
include('Axioms/LCL005-0.ax')
include('Axioms/MSC001-2.ax')
include('Axioms/MSC001-0.ax')
c_in(c_PropLog_Opl_Oop_A_N_{62}(v_p, c_PropLog_Opl_Ofalse, t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) ⇒
c_in(c_PropLog_Opl_Oop_A_N_{62}(v_p, v_q, t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_PropLog_Ofalse_
(c_in(v_p, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) and c_in(c_PropLog_Opl_Oop_A_N_{62}(v_q, c_PropLog_Opl_Ofalse,
c_in(c_PropLog_Opl_Oop_A_N_{62}(c_PropLog_Opl_Oop_A_N_{62}(v_p, v_q, t_a), c_PropLog_Opl_Ofalse, t_a), c_PropLog_Othms(v_H
c_in(c_PropLog_Opl_Oop_A_N_{62}(v_p, v_p, t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_PropLog_Othms_
c_in(v_p, c_PropLog_Othms(v_G, t_a), tc_PropLog_Opl(t_a)) ⇒ c_in(v_p, c_PropLog_Othms(c_union(v_G, v_B, tc_PropLog_Op
c_in(v_p, c_PropLog_Othms(v_G, t_a), tc_PropLog_Opl(t_a)) ⇒ c_in(v_p, c_PropLog_Othms(c_union(v_A, v_G, tc_PropLog_Op
c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) ⇒ c_in(c_PropLog_Opl_Oop_A_N_{62}(v_p, v_q, t_a), c_PropLog_Oth
c_in(v_a, v_tt, t_a) ⇒ c_in(v_a, v_tt, t_a) cnf(cls_conjecture$_0$, negated_conjecture)
c_in(c_PropLog_Opl_Oop_A_N_{62}(c_PropLog_Opl_Ovar(v_a, t_a), c_PropLog_Opl_Ofalse, t_a), c_PropLog_Othms(c_insert(c_Prop
c_in(v_a, v_tt, t_a) cnf(cls_conjecture$_1$, negated_conjecture)
c_in(v_a, v_tt, t_a) ⇒ ¬ c_in(c_PropLog_Opl_Ovar(v_a, t_a), c_PropLog_Othms(c_insert(c_PropLog_Opl_Ovar(v_a, t_a), c_empty
c_in(c_PropLog_Opl_Oop_A_N_{62}(c_PropLog_Opl_Ovar(v_a, t_a), c_PropLog_Opl_Ofalse, t_a), c_PropLog_Othms(c_insert(c_Prop
¬ c_in(c_PropLog_Opl_Ovar(v_a, t_a), c_PropLog_Othms(c_insert(c_PropLog_Opl_Ovar(v_a, t_a), c_emptyset, tc_PropLog_Opl(t

LCL441-2.p Problem about propositional logic
c_in(c_PropLog_Opl_Oop_A_N_{62}(c_PropLog_Opl_Ovar(v_a, t_a), c_PropLog_Opl_Ofalse, t_a), c_PropLog_Othms(c_insert(c_Prop
¬ c_in(c_PropLog_Opl_Ovar(v_a, t_a), c_PropLog_Othms(c_insert(c_PropLog_Opl_Ovar(v_a, t_a), c_emptyset, tc_PropLog_Opl(t
c_in(v_p, v_H, tc_PropLog_Opl(t_a)) ⇒ c_in(v_p, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_PropLog_Oth
c_in(v_x, c_insert(v_x, v_B, t_a), t_a) cnf(cls_Set_OinsertCI$_1$, axiom)

LCL442-2.p Problem about propositional logic
c_PropLog_Oeval(v_tt, v_pl$_1$, t_a) ⇒ c_in(v_pl$_1$, c_PropLog_Othms(c_PropLog_Ohyps(v_pl$_1$, v_tt, t_a), t_a), tc_PropLog_Opl(t_a
c_in(c_PropLog_Opl_Oop_A_N_{62}(v_pl$_1$, c_PropLog_Opl_Ofalse, t_a), c_PropLog_Othms(c_PropLog_Ohyps(v_pl$_1$, v_tt, t_a), t_a),
c_PropLog_Oeval(v_tt, v_pl$_2$, t_a) ⇒ c_in(v_pl$_2$, c_PropLog_Othms(c_PropLog_Ohyps(v_pl$_2$, v_tt, t_a), t_a), tc_PropLog_Opl(t_a

c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_pl$_2$, c_PropLog_Opl_Ofalse, t_a), c_PropLog_Othms(c_PropLog_Ohyps(v_pl$_2$, v_tt, t_a), t_a),
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_pl$_1$, v_pl$_2$, t_a), c_PropLog_Othms(c_union(c_PropLog_Ohyps(v_pl$_1$, v_tt, t_a), c_PropLog_Oh
c_PropLog_Oeval(v_tt, v_pl$_1$, t_a) cnf(cls_conjecture$_4$, negated_conjecture)
c_PropLog_Oeval(v_tt, v_pl$_2$, t_a) ⇒ ¬ c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_pl$_1$, v_pl$_2$, t_a), c_PropLog_Othms(c_union(c_PropLo
(c_in(c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_A_N$_{62}$(v_pl$_1$, v_pl$_2$, t_a), c_PropLog_Opl_Ofalse, t_a), c_PropLog_Othms(
c_PropLog_Oeval(v_tt, v_pl$_2$, t_a) cnf(cls_conjecture$_9$, negated_conjecture)
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, c_PropLog_Opl_Ofalse, t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) ⇒
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_PropLog_Ofalse_
(c_in(v_p, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) and c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_q, c_PropLog_Opl_Ofalse,
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), c_PropLog_Opl_Ofalse, t_a), c_PropLog_Othms(v_H
c_in(v_p, c_PropLog_Othms(v_G, t_a), tc_PropLog_Opl(t_a)) ⇒ c_in(v_p, c_PropLog_Othms(c_union(v_G, v_B, tc_PropLog_Op
c_in(v_p, c_PropLog_Othms(v_G, t_a), tc_PropLog_Opl(t_a)) ⇒ c_in(v_p, c_PropLog_Othms(c_union(v_A, v_G, tc_PropLog_Op
c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) ⇒ c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), c_PropLog_Oth

LCL443-1.p Problem about propositional logic
include('Axioms/LCL005-0.ax')
include('Axioms/MSC001-2.ax')
include('Axioms/MSC001-0.ax')
c_in(v_q, c_PropLog_Othms(c_insert(v_p, v_H, tc_PropLog_Opl(t_a)), t_a), tc_PropLog_Opl(t_a)) ⇒ c_in(c_PropLog_Opl_Oop_
(c_in(v_p, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) and c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), c_PropLog_Ot
c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_PropLog_Othms_OMP$_0$, axiom)
c_in(v_p, c_PropLog_Othms(v_G, t_a), tc_PropLog_Opl(t_a)) ⇒ c_in(v_p, c_PropLog_Othms(c_insert(v_a, v_G, tc_PropLog_Opl
c_in(v_p, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_conjecture$_0$, negated_conjecture)
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_q, c_PropLog_Opl_Ofalse, t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cl
¬ c_in(c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), c_PropLog_Opl_Ofalse, t_a), c_PropLog_Othms(v_

LCL443-2.p Problem about propositional logic
(c_in(v_p, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) and c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), c_PropLog_Ot
c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_PropLog_Othms_OMP$_0$, axiom)
c_in(v_q, c_PropLog_Othms(c_insert(v_p, v_H, tc_PropLog_Opl(t_a)), t_a), tc_PropLog_Opl(t_a)) ⇒ c_in(c_PropLog_Opl_Oop_
c_in(v_p, v_H, tc_PropLog_Opl(t_a)) ⇒ c_in(v_p, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_PropLog_Othr
c_in(v_p, c_PropLog_Othms(v_G, t_a), tc_PropLog_Opl(t_a)) ⇒ c_in(v_p, c_PropLog_Othms(c_insert(v_a, v_G, tc_PropLog_Opl
c_in(v_x, c_insert(v_x, v_B, t_a), t_a) cnf(cls_Set_OinsertCI$_1$, axiom)
c_in(v_p, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_conjecture$_0$, negated_conjecture)
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_q, c_PropLog_Opl_Ofalse, t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cl
¬ c_in(c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), c_PropLog_Opl_Ofalse, t_a), c_PropLog_Othms(v_

LCL444-1.p Problem about propositional logic
include('Axioms/LCL005-0.ax')
include('Axioms/MSC001-2.ax')
include('Axioms/MSC001-0.ax')
c_in(v_q, c_PropLog_Othms(c_insert(v_p, v_H, tc_PropLog_Opl(t_a)), t_a), tc_PropLog_Opl(t_a)) ⇒ c_in(c_PropLog_Opl_Oop_
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, c_PropLog_Opl_Ofalse, t_a), c_Pr
(c_in(v_p, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) and c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), c_PropLog_Ot
c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_PropLog_Othms_OMP$_0$, axiom)
(c_in(v_p, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) and c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), v_H, tc_PropL
c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_PropLog_Othms__H__MP$_0$, axiom)
¬ c_in(c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_

LCL444-2.p Problem about propositional logic
c_in(v_q, c_PropLog_Othms(c_insert(v_p, v_H, tc_PropLog_Opl(t_a)), t_a), tc_PropLog_Opl(t_a)) ⇒ c_in(c_PropLog_Opl_Oop_
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, c_PropLog_Opl_Ofalse, t_a), c_Pr
c_in(v_p, v_H, tc_PropLog_Opl(t_a)) ⇒ c_in(v_p, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_PropLog_Othr
(c_in(v_p, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) and c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), c_PropLog_Ot
c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_PropLog_Othms_OMP$_0$, axiom)
c_in(v_a, v_B, t_a) ⇒ c_in(v_a, c_insert(v_b, v_B, t_a), t_a) cnf(cls_Set_OinsertCI$_0$, axiom)
c_in(v_x, c_insert(v_x, v_B, t_a), t_a) cnf(cls_Set_OinsertCI$_1$, axiom)
¬ c_in(c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_

LCL445-1.p Problem about propositional logic
include('Axioms/LCL005-0.ax')
include('Axioms/MSC001-2.ax')

include('Axioms/MSC001-0.ax')
c_in(v_q, c_PropLog_Othms(c_insert(v_p, v_H, tc_PropLog_Opl(t_a)), t_a), tc_PropLog_Opl(t_a)) ⇒ c_in(c_PropLog_Opl_Oop_
(c_in(v_p, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) and c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), c_PropLog_Ot
c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_PropLog_Othms_OMP$_0$, axiom)
(c_in(v_p, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) and c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), v_H, tc_PropL
c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_PropLog_Othms__H__MP$_0$, axiom)
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_A
c_in(v_q, c_PropLog_Othms(c_insert(v_p, v_H, tc_PropLog_Opl(t_a)), t_a), tc_PropLog_Opl(t_a)) cnf(cls_conjecture$_0$, negate
c_in(v_q, c_PropLog_Othms(c_insert(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, c_PropLog_Opl_Ofalse, t_a), v_H, tc_PropLog_Opl(t_a)), t
¬ c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_conjecture$_2$, negated_conjecture)

LCL445-2.p Problem about propositional logic
c_in(v_q, c_PropLog_Othms(c_insert(v_p, v_H, tc_PropLog_Opl(t_a)), t_a), tc_PropLog_Opl(t_a)) cnf(cls_conjecture$_0$, negate
c_in(v_q, c_PropLog_Othms(c_insert(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, c_PropLog_Opl_Ofalse, t_a), v_H, tc_PropLog_Opl(t_a)), t
¬ c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_conjecture$_2$, negated_conjecture)
c_in(v_q, c_PropLog_Othms(c_insert(v_p, v_H, tc_PropLog_Opl(t_a)), t_a), tc_PropLog_Opl(t_a)) ⇒ c_in(c_PropLog_Opl_Oop_
(c_in(v_p, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) and c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), c_PropLog_Ot
c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_PropLog_Othms_OMP$_0$, axiom)
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_A

LCL446-1.p Problem about propositional logic
include('Axioms/LCL005-0.ax')
include('Axioms/MSC001-2.ax')
include('Axioms/MSC001-0.ax')
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, c_PropLog_Opl_Oop_A_N$_{62}$(v_q, v_p, t_a), t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_C
(c_in(v_p, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) and c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), c_PropLog_Ot
c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_PropLog_Othms_OMP$_0$, axiom)
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, c_PropLog_Opl_Oop_A_N$_{62}$(v_q, v_r, t_a), t_a), c_PropLog_Op
¬ c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_p, t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_conjecture$_0$, ne

LCL446-2.p Problem about propositional logic
¬ c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_p, t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_conjecture$_0$, ne
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, c_PropLog_Opl_Oop_A_N$_{62}$(v_q, v_p, t_a), t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_C
(c_in(v_p, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) and c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), c_PropLog_Ot
c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_PropLog_Othms_OMP$_0$, axiom)
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, c_PropLog_Opl_Oop_A_N$_{62}$(v_q, v_r, t_a), t_a), c_PropLog_Op

LCL447-1.p Problem about propositional logic
include('Axioms/LCL005-0.ax')
include('Axioms/MSC001-2.ax')
include('Axioms/MSC001-0.ax')
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, c_PropLog_Opl_Oop_A_N$_{62}$(v_q, v_p, t_a), t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_C
(c_in(v_p, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) and c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), c_PropLog_Ot
c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_PropLog_Othms_OMP$_0$, axiom)
c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_conjecture$_0$, negated_conjecture)
¬ c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_conjecture$_1$, ne

LCL447-2.p Problem about propositional logic
c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_conjecture$_0$, negated_conjecture)
¬ c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_conjecture$_1$, ne
c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, c_PropLog_Opl_Oop_A_N$_{62}$(v_q, v_p, t_a), t_a), c_PropLog_Othms(v_H, t_a), tc_PropLog_C
(c_in(v_p, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) and c_in(c_PropLog_Opl_Oop_A_N$_{62}$(v_p, v_q, t_a), c_PropLog_Ot
c_in(v_q, c_PropLog_Othms(v_H, t_a), tc_PropLog_Opl(t_a)) cnf(cls_PropLog_Othms_OMP$_0$, axiom)

LCL448+1.p Redundant axiom in Principia axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
op_implies_or fof(principia_op_implies_or, axiom)
op_and fof(principia_op_and, axiom)
op_equiv fof(principia_op_equiv, axiom)
modus_ponens fof(principia_modus_ponens, axiom)
r_1 fof(principia_r$_1$, axiom)
r_2 fof(principia_r$_2$, axiom)

r_3 fof(principia_r$_3$, axiom)
r_5 fof(principia_r$_5$, axiom)
substitution_of_equivalents fof(substitution_of_equivalents, axiom)
r_4 fof(principia_r$_4$, conjecture)

LCL449+1.p Congruence of equiv, to admit substitution of equivalents
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
modus_ponens fof(hilbert_modus_ponens, axiom)
modus_tollens fof(hilbert_modus_tollens, axiom)
implies$_1$ fof(hilbert_implies$_1$, axiom)
implies$_2$ fof(hilbert_implies$_2$, axiom)
implies$_3$ fof(hilbert_implies$_3$, axiom)
and$_1$ fof(hilbert_and$_1$, axiom)
and$_2$ fof(hilbert_and$_2$, axiom)
and$_3$ fof(hilbert_and$_3$, axiom)
or$_1$ fof(hilbert_or$_1$, axiom)
or$_2$ fof(hilbert_or$_2$, axiom)
or$_3$ fof(hilbert_or$_3$, axiom)
equivalence$_1$ fof(hilbert_equivalence$_1$, axiom)
equivalence$_2$ fof(hilbert_equivalence$_2$, axiom)
equivalence$_3$ fof(hilbert_equivalence$_3$, axiom)
($\forall x$: is_a_theorem(equiv(x, x)) and $\forall x, y$: (is_a_theorem(equiv(x, y)) $\Rightarrow$ is_a_theorem(equiv(not(x), not(y)))) and $\forall x_1, x_2, y_1,$
is_a_theorem(equiv(and(x_1, y_1), and(x_2, y_2)))) and $\forall x, y$: ((is_a_theorem(x) and is_a_theorem(equiv(x, y))) $\Rightarrow$ is_a_theorem(
$\forall x, y$: (is_a_theorem(equiv(x, y)) $\Rightarrow$ $x = y$) fof(make_subs_of_equiv, axiom)
$\forall x, y$: (is_a_theorem(equiv(x, y)) $\Rightarrow$ $x = y$) fof(subs_of_equiv, conjecture)

LCL450+1.p Congruence of equiv lemmas, to admit substitution of equivalents
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
modus_ponens fof(hilbert_modus_ponens, axiom)
modus_tollens fof(hilbert_modus_tollens, axiom)
implies$_1$ fof(hilbert_implies$_1$, axiom)
implies$_2$ fof(hilbert_implies$_2$, axiom)
implies$_3$ fof(hilbert_implies$_3$, axiom)
and$_1$ fof(hilbert_and$_1$, axiom)
and$_2$ fof(hilbert_and$_2$, axiom)
and$_3$ fof(hilbert_and$_3$, axiom)
or$_1$ fof(hilbert_or$_1$, axiom)
or$_2$ fof(hilbert_or$_2$, axiom)
or$_3$ fof(hilbert_or$_3$, axiom)
equivalence$_1$ fof(hilbert_equivalence$_1$, axiom)
equivalence$_2$ fof(hilbert_equivalence$_2$, axiom)
equivalence$_3$ fof(hilbert_equivalence$_3$, axiom)
$\forall x$: is_a_theorem(equiv(x, x)) and $\forall x, y$: (is_a_theorem(equiv(x, y)) $\Rightarrow$ is_a_theorem(equiv(not(x), not(y)))) and $\forall x_1, x_2, y_1, y$
is_a_theorem(equiv(and(x_1, y_1), and(x_2, y_2)))) and $\forall x, y$: ((is_a_theorem(x) and is_a_theorem(equiv(x, y))) $\Rightarrow$ is_a_theorem(y

LCL450+2.p Congruence of equiv lemmas, to admit substitution of equivalents
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
modus_ponens fof(hilbert_modus_ponens, axiom)
modus_tollens fof(hilbert_modus_tollens, axiom)

implies_1 fof(hilbert_implies_1, axiom)
implies_2 fof(hilbert_implies_2, axiom)
implies_3 fof(hilbert_implies_3, axiom)
and_1 fof(hilbert_and_1, axiom)
and_2 fof(hilbert_and_2, axiom)
and_3 fof(hilbert_and_3, axiom)
or_1 fof(hilbert_or_1, axiom)
or_2 fof(hilbert_or_2, axiom)
or_3 fof(hilbert_or_3, axiom)
equivalence_1 fof(hilbert_equivalence_1, axiom)
equivalence_2 fof(hilbert_equivalence_2, axiom)
equivalence_3 fof(hilbert_equivalence_3, axiom)
$\forall x$: is_a_theorem(equiv(x, x)) and $\forall x, y$: (is_a_theorem(equiv(x, y)) $\Rightarrow$ is_a_theorem(equiv(not(x), not(y)))) and $\forall x_1, x_2, y$: (is
is_a_theorem(equiv(and(x_1, y), and(x_2, y)))) and $\forall x_1, x_2, y$: (is_a_theorem(equiv(x_1, x_2)) $\Rightarrow$ is_a_theorem(equiv(and(y, x_1), an
is_a_theorem(y)) fof(equiv_congruence, conjecture)

LCL451+1.p Prove Lukasiewicz's cn1 axiom from Hilbert's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+2.ax')
op_or fof(luka_op_or, axiom)
op_implies fof(luka_op_implies, axiom)
op_equiv fof(luka_op_equiv, axiom)
cn_1 fof(luka_cn_1, conjecture)

LCL452+1.p Prove Lukasiewicz's cn2 axiom from Hilbert's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+2.ax')
op_or fof(luka_op_or, axiom)
op_implies fof(luka_op_implies, axiom)
op_equiv fof(luka_op_equiv, axiom)
cn_2 fof(luka_cn_2, conjecture)

LCL453+1.p Prove Lukasiewicz's cn3 axiom from Hilbert's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+2.ax')
op_or fof(luka_op_or, axiom)
op_implies fof(luka_op_implies, axiom)
op_equiv fof(luka_op_equiv, axiom)
cn_3 fof(luka_cn_3, conjecture)

LCL454+1.p Prove Principia's r1 axiom from Hilbert's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+2.ax')
op_implies_or fof(principia_op_implies_or, axiom)
op_and fof(principia_op_and, axiom)
op_equiv fof(principia_op_equiv, axiom)
r_1 fof(principia_r_1, conjecture)

LCL455+1.p Prove Principia's r2 axiom from Hilbert's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+2.ax')
op_implies_or fof(principia_op_implies_or, axiom)
op_and fof(principia_op_and, axiom)
op_equiv fof(principia_op_equiv, axiom)
r_2 fof(principia_r_2, conjecture)

LCL456+1.p Prove Principia's r3 axiom from Hilbert's axiomatization
include('Axioms/LCL006+0.ax')

include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+2.ax')
op_implies_or   fof(principia_op_implies_or, axiom)
op_and   fof(principia_op_and, axiom)
op_equiv   fof(principia_op_equiv, axiom)
r_3 fof(principia_r_3, conjecture)

LCL457+1.p Prove Principia's r4 axiom from Hilbert's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+2.ax')
op_implies_or   fof(principia_op_implies_or, axiom)
op_and   fof(principia_op_and, axiom)
op_equiv   fof(principia_op_equiv, axiom)
r_4 fof(principia_r_4, conjecture)

LCL458+1.p Prove Principia's r5 axiom from Hilbert's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+2.ax')
op_implies_or   fof(principia_op_implies_or, axiom)
op_and   fof(principia_op_and, axiom)
op_equiv   fof(principia_op_equiv, axiom)
r_5 fof(principia_r_5, conjecture)

LCL459+1.p Prove Rosser's kn1 axiom from Hilbert's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+2.ax')
op_or   fof(rosser_op_or, axiom)
op_implies_and   fof(rosser_op_implies_and, axiom)
op_equiv   fof(rosser_op_equiv, axiom)
kn_1 fof(rosser_kn_1, conjecture)

LCL460+1.p Prove Rosser's kn2 axiom from Hilbert's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+2.ax')
op_or   fof(rosser_op_or, axiom)
op_implies_and   fof(rosser_op_implies_and, axiom)
op_equiv   fof(rosser_op_equiv, axiom)
kn_2 fof(rosser_kn_2, conjecture)

LCL461+1.p Prove Rosser's kn3 axiom from Hilbert's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+2.ax')
op_or   fof(rosser_op_or, axiom)
op_implies_and   fof(rosser_op_implies_and, axiom)
op_equiv   fof(rosser_op_equiv, axiom)
kn_3 fof(rosser_kn_3, conjecture)

LCL462+1.p Prove Hilbert's modus_tollens axiom from Lukasiewicz's system
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+3.ax')
op_or   fof(hilbert_op_or, axiom)
op_implies_and   fof(hilbert_op_implies_and, axiom)
op_equiv   fof(hilbert_op_equiv, axiom)
modus_tollens   fof(hilbert_modus_tollens, conjecture)

LCL463+1.p Prove Hilbert's implies_1 axiom from Lukasiewicz's axiomatization
include('Axioms/LCL006+0.ax')

include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+3.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
implies$_1$ fof(hilbert_implies$_1$, conjecture)

LCL464+1.p Prove Hilbert's implies_2 axiom from Lukasiewicz's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+3.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
implies$_2$ fof(hilbert_implies$_2$, conjecture)

LCL465+1.p Prove Hilbert's implies_3 axiom from Lukasiewicz's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+3.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
implies$_3$ fof(hilbert_implies$_3$, conjecture)

LCL466+1.p Prove Hilbert's and_1 axiom from Lukasiewicz's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+3.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
and$_1$ fof(hilbert_and$_1$, conjecture)

LCL467+1.p Prove Hilbert's and_2 axiom from Lukasiewicz's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+3.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
and$_2$ fof(hilbert_and$_2$, conjecture)

LCL468+1.p Prove Hilbert's and_3 axiom from Lukasiewicz's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+3.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
and$_3$ fof(hilbert_and$_3$, conjecture)

LCL469+1.p Prove Hilbert's or_1 axiom from Lukasiewicz's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+3.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
or$_1$ fof(hilbert_or$_1$, conjecture)

LCL470+1.p Prove Hilbert's or_2 axiom from Lukasiewicz's axiomatization
include('Axioms/LCL006+0.ax')

include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+3.ax')
op_or   fof(hilbert_op_or, axiom)
op_implies_and   fof(hilbert_op_implies_and, axiom)
op_equiv   fof(hilbert_op_equiv, axiom)
or_2 fof(hilbert_or_2, conjecture)

LCL471+1.p Prove Hilbert's or_3 axiom from Lukasiewicz's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+3.ax')
op_or   fof(hilbert_op_or, axiom)
op_implies_and   fof(hilbert_op_implies_and, axiom)
op_equiv   fof(hilbert_op_equiv, axiom)
or_3 fof(hilbert_or_3, conjecture)

LCL472+1.p Prove Hilbert's equivalence_1 axiom from Lukasiewicz's system
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+3.ax')
op_or   fof(hilbert_op_or, axiom)
op_implies_and   fof(hilbert_op_implies_and, axiom)
op_equiv   fof(hilbert_op_equiv, axiom)
equivalence_1 fof(hilbert_equivalence_1, conjecture)

LCL473+1.p Prove Hilbert's equivalence_2 axiom from Lukasiewicz's system
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+3.ax')
op_or   fof(hilbert_op_or, axiom)
op_implies_and   fof(hilbert_op_implies_and, axiom)
op_equiv   fof(hilbert_op_equiv, axiom)
equivalence_2 fof(hilbert_equivalence_2, conjecture)

LCL474+1.p Prove Hilbert's equivalence_3 axiom from Lukasiewicz's system
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+3.ax')
op_or   fof(hilbert_op_or, axiom)
op_implies_and   fof(hilbert_op_implies_and, axiom)
op_equiv   fof(hilbert_op_equiv, axiom)
equivalence_3 fof(hilbert_equivalence_3, conjecture)

LCL475+1.p Prove Principia's r1 axiom from Lukasiewicz's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+3.ax')
op_implies_or   fof(principia_op_implies_or, axiom)
op_and   fof(principia_op_and, axiom)
op_equiv   fof(principia_op_equiv, axiom)
r_1 fof(principia_r_1, conjecture)

LCL476+1.p Prove Principia's r2 axiom from Lukasiewicz's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+3.ax')
op_implies_or   fof(principia_op_implies_or, axiom)
op_and   fof(principia_op_and, axiom)
op_equiv   fof(principia_op_equiv, axiom)
r_2 fof(principia_r_2, conjecture)

LCL477+1.p Prove Principia's r3 axiom from Lukasiewicz's axiomatization
include('Axioms/LCL006+0.ax')

include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+3.ax')
op_implies_or   fof(principia_op_implies_or, axiom)
op_and   fof(principia_op_and, axiom)
op_equiv   fof(principia_op_equiv, axiom)
r_3 fof(principia_r$_3$, conjecture)

LCL478+1.p Prove Principia's r4 axiom from Lukasiewicz's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+3.ax')
op_implies_or   fof(principia_op_implies_or, axiom)
op_and   fof(principia_op_and, axiom)
op_equiv   fof(principia_op_equiv, axiom)
r_4 fof(principia_r$_4$, conjecture)

LCL479+1.p Prove Principia's r5 axiom from Lukasiewicz's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+3.ax')
op_implies_or   fof(principia_op_implies_or, axiom)
op_and   fof(principia_op_and, axiom)
op_equiv   fof(principia_op_equiv, axiom)
r_5 fof(principia_r$_5$, conjecture)

LCL480+1.p Prove Rosser's kn1 axiom from Lukasiewicz's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+3.ax')
op_or   fof(rosser_op_or, axiom)
op_implies_and   fof(rosser_op_implies_and, axiom)
op_equiv   fof(rosser_op_equiv, axiom)
kn$_1$ fof(rosser_kn$_1$, conjecture)

LCL481+1.p Prove Rosser's kn2 axiom from Lukasiewicz's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+3.ax')
op_or   fof(rosser_op_or, axiom)
op_implies_and   fof(rosser_op_implies_and, axiom)
op_equiv   fof(rosser_op_equiv, axiom)
kn$_2$ fof(rosser_kn$_2$, conjecture)

LCL482+1.p Prove Rosser's kn3 axiom from Lukasiewicz's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+3.ax')
op_or   fof(rosser_op_or, axiom)
op_implies_and   fof(rosser_op_implies_and, axiom)
op_equiv   fof(rosser_op_equiv, axiom)
kn$_3$ fof(rosser_kn$_3$, conjecture)

LCL483+1.p Prove Hilbert's modus_tollens axiom from Principia's system
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+4.ax')
op_or   fof(hilbert_op_or, axiom)
op_implies_and   fof(hilbert_op_implies_and, axiom)
op_equiv   fof(hilbert_op_equiv, axiom)
modus_tollens   fof(hilbert_modus_tollens, conjecture)

LCL484+1.p Prove Hilbert's implies_1 axiom from Principia's axiomatization
include('Axioms/LCL006+0.ax')

include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+4.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
implies_1 fof(hilbert_implies_1, conjecture)

LCL485+1.p Prove Hilbert's implies_2 axiom from Principia's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+4.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
implies_2 fof(hilbert_implies_2, conjecture)

LCL486+1.p Prove Hilbert's implies_3 axiom from Principia's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+4.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
implies_3 fof(hilbert_implies_3, conjecture)

LCL487+1.p Prove Hilbert's and_1 axiom from Principia's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+4.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
and_1 fof(hilbert_and_1, conjecture)

LCL488+1.p Prove Hilbert's and_2 axiom from Principia's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+4.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
and_2 fof(hilbert_and_2, conjecture)

LCL489+1.p Prove Hilbert's and_3 axiom from Principia's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+4.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
and_3 fof(hilbert_and_3, conjecture)

LCL490+1.p Prove Hilbert's or_1 axiom from Principia's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+4.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
or_1 fof(hilbert_or_1, conjecture)

LCL491+1.p Prove Hilbert's or_2 axiom from Principia's axiomatization
include('Axioms/LCL006+0.ax')

include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+4.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
or_2 fof(hilbert_or_2, conjecture)

LCL492+1.p Prove Hilbert's or_3 axiom from Principia's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+4.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
or_3 fof(hilbert_or_3, conjecture)

LCL493+1.p Prove Hilbert's equivalence_1 axiom from Principia's system
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+4.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
$equivalence_1$ fof(hilbert_$equivalence_1$, conjecture)

LCL494+1.p Prove Hilbert's equivalence_2 axiom from Principia's system
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+4.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
$equivalence_2$ fof(hilbert_$equivalence_2$, conjecture)

LCL495+1.p Prove Hilbert's equivalence_3 axiom from Principia's system
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+4.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
$equivalence_3$ fof(hilbert_$equivalence_3$, conjecture)

LCL496+1.p Prove Lukasiewicz's cn1 axiom from Principia's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+4.ax')
op_or fof(luka_op_or, axiom)
op_implies fof(luka_op_implies, axiom)
op_equiv fof(luka_op_equiv, axiom)
cn_1 fof(luka_cn_1, conjecture)

LCL497+1.p Prove Lukasiewicz's cn2 axiom from Principia's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+4.ax')
op_or fof(luka_op_or, axiom)
op_implies fof(luka_op_implies, axiom)
op_equiv fof(luka_op_equiv, axiom)
cn_2 fof(luka_cn_2, conjecture)

LCL498+1.p Prove Lukasiewicz's cn3 axiom from Principia's axiomatization
include('Axioms/LCL006+0.ax')

include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+4.ax')
op_or fof(luka_op_or, axiom)
op_implies fof(luka_op_implies, axiom)
op_equiv fof(luka_op_equiv, axiom)
cn_3 fof(luka_cn_3, conjecture)

LCL499+1.p Prove Rosser's kn1 axiom from Principia's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+4.ax')
op_or fof(rosser_op_or, axiom)
op_implies_and fof(rosser_op_implies_and, axiom)
op_equiv fof(rosser_op_equiv, axiom)
kn_1 fof(rosser_kn_1, conjecture)

LCL500+1.p Prove Rosser's kn2 axiom from Principia's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+4.ax')
op_or fof(rosser_op_or, axiom)
op_implies_and fof(rosser_op_implies_and, axiom)
op_equiv fof(rosser_op_equiv, axiom)
kn_2 fof(rosser_kn_2, conjecture)

LCL501+1.p Prove Rosser's kn3 axiom from Principia's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+4.ax')
op_or fof(rosser_op_or, axiom)
op_implies_and fof(rosser_op_implies_and, axiom)
op_equiv fof(rosser_op_equiv, axiom)
kn_3 fof(rosser_kn_3, conjecture)

LCL502+1.p Prove Hilbert's modus_tollens axiom from Rosser's system
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+5.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
modus_tollens fof(hilbert_modus_tollens, conjecture)

LCL503+1.p Prove Hilbert's implies_1 axiom from Rosser's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+5.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
$implies_1$ fof(hilbert_$implies_1$, conjecture)

LCL504+1.p Prove Hilbert's implies_2 axiom from Rosser's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+5.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
$implies_2$ fof(hilbert_$implies_2$, conjecture)

LCL505+1.p Prove Hilbert's implies_3 axiom from Rosser's axiomatization
include('Axioms/LCL006+0.ax')

include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+5.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
implies_3 fof(hilbert_implies_3, conjecture)

LCL506+1.p Prove Hilbert's and_1 axiom from Rosser's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+5.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
and_1 fof(hilbert_and_1, conjecture)

LCL507+1.p Prove Hilbert's and_2 axiom from Rosser's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+5.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
and_2 fof(hilbert_and_2, conjecture)

LCL508+1.p Prove Hilbert's and_3 axiom from Rosser's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+5.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
and_3 fof(hilbert_and_3, conjecture)

LCL509+1.p Prove Hilbert's or_1 axiom from Rosser's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+5.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
or_1 fof(hilbert_or_1, conjecture)

LCL510+1.p Prove Hilbert's or_2 axiom from Rosser's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+5.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
or_2 fof(hilbert_or_2, conjecture)

LCL511+1.p Prove Hilbert's or_3 axiom from Rosser's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+5.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
or_3 fof(hilbert_or_3, conjecture)

LCL512+1.p Prove Hilbert's equivalence_1 axiom from Rosser's system
include('Axioms/LCL006+0.ax')

include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+5.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
equivalence_1 fof($\text{hilbert\_equivalence}_1$, conjecture)

LCL513+1.p Prove Hilbert's equivalence_2 axiom from Rosser's system
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+5.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
equivalence_2 fof($\text{hilbert\_equivalence}_2$, conjecture)

LCL514+1.p Prove Hilbert's equivalence_3 axiom from Rosser's system
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+5.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
equivalence_3 fof($\text{hilbert\_equivalence}_3$, conjecture)

LCL515+1.p Prove Lukasiewicz's cn1 axiom from Rosser's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+5.ax')
op_or fof(luka_op_or, axiom)
op_implies fof(luka_op_implies, axiom)
op_equiv fof(luka_op_equiv, axiom)
cn_1 fof(luka_cn_1, conjecture)

LCL516+1.p Prove Lukasiewicz's cn2 axiom from Rosser's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+5.ax')
op_or fof(luka_op_or, axiom)
op_implies fof(luka_op_implies, axiom)
op_equiv fof(luka_op_equiv, axiom)
cn_2 fof(luka_cn_2, conjecture)

LCL517+1.p Prove Lukasiewicz's cn3 axiom from Rosser's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+5.ax')
op_or fof(luka_op_or, axiom)
op_implies fof(luka_op_implies, axiom)
op_equiv fof(luka_op_equiv, axiom)
cn_3 fof(luka_cn_3, conjecture)

LCL518+1.p Prove Principia's r1 axiom from Rosser's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+5.ax')
op_implies_or fof(principia_op_implies_or, axiom)
op_and fof(principia_op_and, axiom)
op_equiv fof(principia_op_equiv, axiom)
r_1 fof(principia_r_1, conjecture)

LCL519+1.p Prove Principia's r2 axiom from Rosser's axiomatization
include('Axioms/LCL006+0.ax')

include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+5.ax')
op_implies_or fof(principia_op_implies_or, axiom)
op_and fof(principia_op_and, axiom)
op_equiv fof(principia_op_equiv, axiom)
r_2 fof(principia_r$_2$, conjecture)

LCL520+1.p Prove Principia's r3 axiom from Rosser's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+5.ax')
op_implies_or fof(principia_op_implies_or, axiom)
op_and fof(principia_op_and, axiom)
op_equiv fof(principia_op_equiv, axiom)
r_3 fof(principia_r$_3$, conjecture)

LCL521+1.p Prove Principia's r4 axiom from Rosser's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+5.ax')
op_implies_or fof(principia_op_implies_or, axiom)
op_and fof(principia_op_and, axiom)
op_equiv fof(principia_op_equiv, axiom)
r_4 fof(principia_r$_4$, conjecture)

LCL522+1.p Prove Principia's r5 axiom from Rosser's axiomatization
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+5.ax')
op_implies_or fof(principia_op_implies_or, axiom)
op_and fof(principia_op_and, axiom)
op_equiv fof(principia_op_equiv, axiom)
r_5 fof(principia_r$_5$, conjecture)

LCL523+1.p Prove axiom 4 from KM5 axiomatization of S5
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+2.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+2.ax')
axiom$_4$ fof(km4b_axiom$_4$, conjecture)

LCL524+1.p Prove axiom B from KM5 axiomatization of S5
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+2.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+2.ax')
axiom_B fof(km4b_axiom_B, conjecture)

LCL525+1.p Prove strict implies modus ponens from KM5 axiomatization of S5
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+2.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+2.ax')
op_possibly fof(s1_0_op_possibly, axiom)
op_or fof(s1_0_op_or, axiom)
op_implies fof(s1_0_op_implies, axiom)
op_strict_implies fof(s1_0_op_strict_implies, axiom)

op_equiv fof(s1_0_op_equiv, axiom)
op_strict_equiv fof(s1_0_op_strict_equiv, axiom)
modus_ponens_strict_implies fof(s1_0_modus_ponens_strict_implies, conjecture)

LCL526+1.p Prove SSE from KM5 axiomatization of S5
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+2.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+2.ax')
op_possibly fof(s1_0_op_possibly, axiom)
op_or fof(s1_0_op_or, axiom)
op_implies fof(s1_0_op_implies, axiom)
op_strict_implies fof(s1_0_op_strict_implies, axiom)
op_equiv fof(s1_0_op_equiv, axiom)
op_strict_equiv fof(s1_0_op_strict_equiv, axiom)
substitution_strict_equiv fof(s1_0_substitution_strict_equiv, conjecture)

LCL527+1.p Prove adjunction from KM5 axiomatization of S5
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+2.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+2.ax')
op_possibly fof(s1_0_op_possibly, axiom)
op_or fof(s1_0_op_or, axiom)
op_implies fof(s1_0_op_implies, axiom)
op_strict_implies fof(s1_0_op_strict_implies, axiom)
op_equiv fof(s1_0_op_equiv, axiom)
op_strict_equiv fof(s1_0_op_strict_equiv, axiom)
adjunction fof(s1_0_adjunction, conjecture)

LCL528+1.p Prove axiom m1 from KM5 axiomatization of S5
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+2.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+2.ax')
op_possibly fof(s1_0_op_possibly, axiom)
op_or fof(s1_0_op_or, axiom)
op_implies fof(s1_0_op_implies, axiom)
op_strict_implies fof(s1_0_op_strict_implies, axiom)
op_equiv fof(s1_0_op_equiv, axiom)
op_strict_equiv fof(s1_0_op_strict_equiv, axiom)
axiom_m_1 fof(s1_0_axiom_m_1, conjecture)

LCL529+1.p Prove axiom m2 from KM5 axiomatization of S5
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+2.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+2.ax')
op_possibly fof(s1_0_op_possibly, axiom)
op_or fof(s1_0_op_or, axiom)
op_implies fof(s1_0_op_implies, axiom)
op_strict_implies fof(s1_0_op_strict_implies, axiom)
op_equiv fof(s1_0_op_equiv, axiom)

op_strict_equiv   fof(s1_0_op_strict_equiv, axiom)
axiom_m_2 fof(s1_0_axiom_m_2, conjecture)

LCL530+1.p Prove axiom m3 from KM5 axiomatization of S5
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+2.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+2.ax')
op_possibly   fof(s1_0_op_possibly, axiom)
op_or   fof(s1_0_op_or, axiom)
op_implies   fof(s1_0_op_implies, axiom)
op_strict_implies   fof(s1_0_op_strict_implies, axiom)
op_equiv   fof(s1_0_op_equiv, axiom)
op_strict_equiv   fof(s1_0_op_strict_equiv, axiom)
axiom_m_3 fof(s1_0_axiom_m_3, conjecture)

LCL531+1.p Prove axiom m4 from KM5 axiomatization of S5
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+2.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+2.ax')
op_possibly   fof(s1_0_op_possibly, axiom)
op_or   fof(s1_0_op_or, axiom)
op_implies   fof(s1_0_op_implies, axiom)
op_strict_implies   fof(s1_0_op_strict_implies, axiom)
op_equiv   fof(s1_0_op_equiv, axiom)
op_strict_equiv   fof(s1_0_op_strict_equiv, axiom)
axiom_m_4 fof(s1_0_axiom_m_4, conjecture)

LCL532+1.p Prove axiom m5 from KM5 axiomatization of S5
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+2.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+2.ax')
op_possibly   fof(s1_0_op_possibly, axiom)
op_or   fof(s1_0_op_or, axiom)
op_implies   fof(s1_0_op_implies, axiom)
op_strict_implies   fof(s1_0_op_strict_implies, axiom)
op_equiv   fof(s1_0_op_equiv, axiom)
op_strict_equiv   fof(s1_0_op_strict_equiv, axiom)
axiom_m_5 fof(s1_0_axiom_m_5, conjecture)

LCL533+1.p Prove axiom m6 from KM5 axiomatization of S5
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+2.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+2.ax')
op_possibly   fof(s1_0_op_possibly, axiom)
op_or   fof(s1_0_op_or, axiom)
op_implies   fof(s1_0_op_implies, axiom)
op_strict_implies   fof(s1_0_op_strict_implies, axiom)
op_equiv   fof(s1_0_op_equiv, axiom)
op_strict_equiv   fof(s1_0_op_strict_equiv, axiom)

axiom_m_6 fof(s1_0_m6s3m9b_axiom_m_6, conjecture)

LCL534+1.p Prove axiom s3 from KM5 axiomatization of S5
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+2.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+2.ax')
op_possibly fof(s1_0_op_possibly, axiom)
op_or fof(s1_0_op_or, axiom)
op_implies fof(s1_0_op_implies, axiom)
op_strict_implies fof(s1_0_op_strict_implies, axiom)
op_equiv fof(s1_0_op_equiv, axiom)
op_strict_equiv fof(s1_0_op_strict_equiv, axiom)
axiom_s_3 fof(s1_0_m6s3m9b_axiom_s_3, conjecture)

LCL535+1.p Prove axiom m9 from KM5 axiomatization of S5
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+2.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+2.ax')
op_possibly fof(s1_0_op_possibly, axiom)
op_or fof(s1_0_op_or, axiom)
op_implies fof(s1_0_op_implies, axiom)
op_strict_implies fof(s1_0_op_strict_implies, axiom)
op_equiv fof(s1_0_op_equiv, axiom)
op_strict_equiv fof(s1_0_op_strict_equiv, axiom)
axiom_m_9 fof(s1_0_m6s3m9b_axiom_m_9, conjecture)

LCL536+1.p Prove axiom m10 from KM5 axiomatization of S5
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+2.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+2.ax')
op_possibly fof(s1_0_op_possibly, axiom)
op_or fof(s1_0_op_or, axiom)
op_implies fof(s1_0_op_implies, axiom)
op_strict_implies fof(s1_0_op_strict_implies, axiom)
op_equiv fof(s1_0_op_equiv, axiom)
op_strict_equiv fof(s1_0_op_strict_equiv, axiom)
axiom_m_{10} fof(s1_0_m10_axiom_m_{10}, conjecture)

LCL537+1.p Prove axiom 5 from KM4B axiomatization of S5
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+2.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+3.ax')
$axiom_5$ fof(km5_$axiom_5$, conjecture)

LCL538+1.p Prove strict implies modus ponens from KM4B axiomatization of S5
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+2.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')

include('Axioms/LCL007+3.ax')
op_possibly fof(s1_0_op_possibly, axiom)
op_or fof(s1_0_op_or, axiom)
op_implies fof(s1_0_op_implies, axiom)
op_strict_implies fof(s1_0_op_strict_implies, axiom)
op_equiv fof(s1_0_op_equiv, axiom)
op_strict_equiv fof(s1_0_op_strict_equiv, axiom)
modus_ponens_strict_implies fof(s1_0_modus_ponens_strict_implies, conjecture)

LCL539+1.p Prove SSE from KM4B axiomatization of S5
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+2.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+3.ax')
op_possibly fof(s1_0_op_possibly, axiom)
op_or fof(s1_0_op_or, axiom)
op_implies fof(s1_0_op_implies, axiom)
op_strict_implies fof(s1_0_op_strict_implies, axiom)
op_equiv fof(s1_0_op_equiv, axiom)
op_strict_equiv fof(s1_0_op_strict_equiv, axiom)
substitution_strict_equiv fof(s1_0_substitution_strict_equiv, conjecture)

LCL540+1.p Prove adjunction from KM4B axiomatization of S5
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+2.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+3.ax')
op_possibly fof(s1_0_op_possibly, axiom)
op_or fof(s1_0_op_or, axiom)
op_implies fof(s1_0_op_implies, axiom)
op_strict_implies fof(s1_0_op_strict_implies, axiom)
op_equiv fof(s1_0_op_equiv, axiom)
op_strict_equiv fof(s1_0_op_strict_equiv, axiom)
adjunction fof(s1_0_adjunction, conjecture)

LCL541+1.p Prove axiom m1 from KM4B axiomatization of S5
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+2.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+3.ax')
op_possibly fof(s1_0_op_possibly, axiom)
op_or fof(s1_0_op_or, axiom)
op_implies fof(s1_0_op_implies, axiom)
op_strict_implies fof(s1_0_op_strict_implies, axiom)
op_equiv fof(s1_0_op_equiv, axiom)
op_strict_equiv fof(s1_0_op_strict_equiv, axiom)
axiom_m_1 fof(s1_0_axiom_m_1, conjecture)

LCL542+1.p Prove axiom m2 from KM4B axiomatization of S5
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+2.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+3.ax')

op_possibly fof(s1_0_op_possibly, axiom)
op_or fof(s1_0_op_or, axiom)
op_implies fof(s1_0_op_implies, axiom)
op_strict_implies fof(s1_0_op_strict_implies, axiom)
op_equiv fof(s1_0_op_equiv, axiom)
op_strict_equiv fof(s1_0_op_strict_equiv, axiom)
axiom_m_2 fof(s1_0_axiom_m_2, conjecture)

LCL543+1.p Prove axiom m3 from KM4B axiomatization of S5
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+2.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+3.ax')
op_possibly fof(s1_0_op_possibly, axiom)
op_or fof(s1_0_op_or, axiom)
op_implies fof(s1_0_op_implies, axiom)
op_strict_implies fof(s1_0_op_strict_implies, axiom)
op_equiv fof(s1_0_op_equiv, axiom)
op_strict_equiv fof(s1_0_op_strict_equiv, axiom)
axiom_m_3 fof(s1_0_axiom_m_3, conjecture)

LCL544+1.p Prove axiom m4 from KM4B axiomatization of S5
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+2.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+3.ax')
op_possibly fof(s1_0_op_possibly, axiom)
op_or fof(s1_0_op_or, axiom)
op_implies fof(s1_0_op_implies, axiom)
op_strict_implies fof(s1_0_op_strict_implies, axiom)
op_equiv fof(s1_0_op_equiv, axiom)
op_strict_equiv fof(s1_0_op_strict_equiv, axiom)
axiom_m_4 fof(s1_0_axiom_m_4, conjecture)

LCL545+1.p Prove axiom m5 from KM4B axiomatization of S5
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+2.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+3.ax')
op_possibly fof(s1_0_op_possibly, axiom)
op_or fof(s1_0_op_or, axiom)
op_implies fof(s1_0_op_implies, axiom)
op_strict_implies fof(s1_0_op_strict_implies, axiom)
op_equiv fof(s1_0_op_equiv, axiom)
op_strict_equiv fof(s1_0_op_strict_equiv, axiom)
axiom_m_5 fof(s1_0_axiom_m_5, conjecture)

LCL546+1.p Prove axiom m6 from KM4B axiomatization of S5
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+2.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+3.ax')
op_possibly fof(s1_0_op_possibly, axiom)

op_or fof(s1_0_op_or, axiom)
op_implies fof(s1_0_op_implies, axiom)
op_strict_implies fof(s1_0_op_strict_implies, axiom)
op_equiv fof(s1_0_op_equiv, axiom)
op_strict_equiv fof(s1_0_op_strict_equiv, axiom)
axiom_m_6 fof(s1_0_m6s3m9b_axiom_m_6, conjecture)

LCL547+1.p Prove axiom s3 from KM4B axiomatization of S5
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+2.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+3.ax')
op_possibly fof(s1_0_op_possibly, axiom)
op_or fof(s1_0_op_or, axiom)
op_implies fof(s1_0_op_implies, axiom)
op_strict_implies fof(s1_0_op_strict_implies, axiom)
op_equiv fof(s1_0_op_equiv, axiom)
op_strict_equiv fof(s1_0_op_strict_equiv, axiom)
axiom_s_3 fof(s1_0_m6s3m9b_axiom_s_3, conjecture)

LCL548+1.p Prove axiom m9 from KM4B axiomatization of S5
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+2.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+3.ax')
op_possibly fof(s1_0_op_possibly, axiom)
op_or fof(s1_0_op_or, axiom)
op_implies fof(s1_0_op_implies, axiom)
op_strict_implies fof(s1_0_op_strict_implies, axiom)
op_equiv fof(s1_0_op_equiv, axiom)
op_strict_equiv fof(s1_0_op_strict_equiv, axiom)
axiom_m_9 fof(s1_0_m6s3m9b_axiom_m_9, conjecture)

LCL549+1.p Prove axiom m10 from KM4B axiomatization of S5
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+2.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+3.ax')
op_possibly fof(s1_0_op_possibly, axiom)
op_or fof(s1_0_op_or, axiom)
op_implies fof(s1_0_op_implies, axiom)
op_strict_implies fof(s1_0_op_strict_implies, axiom)
op_equiv fof(s1_0_op_equiv, axiom)
op_strict_equiv fof(s1_0_op_strict_equiv, axiom)
axiom_m_{10} fof(s1_0_m10_axiom_m_{10}, conjecture)

LCL550+1.p Prove Hilbert's modus ponens rule from the S1-0 system
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+4.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)

substitution_of_equivalents fof(substitution_of_equivalents, axiom)
modus_ponens fof(hilbert_modus_ponens, conjecture)

LCL551+1.p Prove Hilbert's modus_tollens axiom from the S1-0 system
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+4.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
substitution_of_equivalents fof(substitution_of_equivalents, axiom)
modus_tollens fof(hilbert_modus_tollens, conjecture)

LCL552+1.p Prove Hilbert's implies_1 axiom from the S1-0 system
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+4.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
substitution_of_equivalents fof(substitution_of_equivalents, axiom)
implies_1 fof(hilbert_implies_1, conjecture)

LCL553+1.p Prove Hilbert's implies_2 axiom from the S1-0 system
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+4.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
substitution_of_equivalents fof(substitution_of_equivalents, axiom)
implies_2 fof(hilbert_implies_2, conjecture)

LCL554+1.p Prove Hilbert's implies_3 axiom from the S1-0 system
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+4.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
substitution_of_equivalents fof(substitution_of_equivalents, axiom)
implies_3 fof(hilbert_implies_3, conjecture)

LCL555+1.p Prove Hilbert's and_1 axiom from the S1-0 system
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+4.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
substitution_of_equivalents fof(substitution_of_equivalents, axiom)
and_1 fof(hilbert_and_1, conjecture)

LCL556+1.p Prove Hilbert's and_2 axiom from the S1-0 system
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+4.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
substitution_of_equivalents fof(substitution_of_equivalents, axiom)
and_2 fof(hilbert_and_2, conjecture)

LCL557+1.p Prove Hilbert's and_3 axiom from the S1-0 system
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+4.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
substitution_of_equivalents fof(substitution_of_equivalents, axiom)
and_3 fof(hilbert_and_3, conjecture)

LCL558+1.p Prove Hilbert's or_1 axiom from the S1-0 system
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+4.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
substitution_of_equivalents fof(substitution_of_equivalents, axiom)
or_1 fof(hilbert_or_1, conjecture)

LCL559+1.p Prove Hilbert's or_2 axiom from the S1-0 system
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+4.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
substitution_of_equivalents fof(substitution_of_equivalents, axiom)
or_2 fof(hilbert_or_2, conjecture)

LCL560+1.p Prove Hilbert's or_3 axiom from the S1-0 system
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+4.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
substitution_of_equivalents fof(substitution_of_equivalents, axiom)
or_3 fof(hilbert_or_3, conjecture)

LCL561+1.p Prove Hilbert's equivalence_1 axiom from the S1-0 system
include('Axioms/LCL006+0.ax')

include('Axioms/LCL006+1.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+4.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
substitution_of_equivalents fof(substitution_of_equivalents, axiom)
equivalence_1 fof($\text{hilbert\_equivalence}_1$, conjecture)

LCL562+1.p Prove Hilbert's equivalence_2 axiom from the S1-0 system
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+4.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
substitution_of_equivalents fof(substitution_of_equivalents, axiom)
equivalence_2 fof($\text{hilbert\_equivalence}_2$, conjecture)

LCL563+1.p Prove Hilbert's equivalence_3 axiom from the S1-0 system
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+4.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
substitution_of_equivalents fof(substitution_of_equivalents, axiom)
equivalence_3 fof($\text{hilbert\_equivalence}_3$, conjecture)

LCL564+1.p Prove axiom K from the S1-0M6S3M9B axiomatization of S5
include('Axioms/LCL006+1.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+4.ax')
include('Axioms/LCL007+5.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
substitution_of_equivalents fof(substitution_of_equivalents, axiom)
axiom_K fof(km5_axiom_K, conjecture)

LCL565+1.p Prove necessitation from the S1-0M6S3M9B axiomatization of S5
include('Axioms/LCL006+1.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+4.ax')
include('Axioms/LCL007+5.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
substitution_of_equivalents fof(substitution_of_equivalents, axiom)
necessitation fof(km5_necessitation, conjecture)

LCL566+1.p Prove axiom M from the S1-0M6S3M9B axiomatization of S5
include('Axioms/LCL006+1.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')

include('Axioms/LCL007+4.ax')
include('Axioms/LCL007+5.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
substitution_of_equivalents fof(substitution_of_equivalents, axiom)
axiom_M fof(km5_axiom_M, conjecture)

LCL567+1.p Prove axiom 5 from the S1-0M6S3M9B axiomatization of S5
include('Axioms/LCL006+1.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+4.ax')
include('Axioms/LCL007+5.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
substitution_of_equivalents fof(substitution_of_equivalents, axiom)
$axiom_5$ fof(km5_$axiom_5$, conjecture)

LCL568+1.p Prove axiom 4 from the S1-0M6S3M9B axiomatization of S5
include('Axioms/LCL006+1.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+4.ax')
include('Axioms/LCL007+5.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
substitution_of_equivalents fof(substitution_of_equivalents, axiom)
$axiom_4$ fof(km4b_$axiom_4$, conjecture)

LCL569+1.p Prove axiom m10 from the S1-0M6S3M9B axiomatization of S5
include('Axioms/LCL006+1.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+4.ax')
include('Axioms/LCL007+5.ax')
axiom_m_{10} fof(s1_0_m10_axiom_m_{10}, conjecture)

LCL570+1.p Prove axiom K from the S1-0M10 axiomatization of S5
include('Axioms/LCL006+1.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+4.ax')
include('Axioms/LCL007+6.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
substitution_of_equivalents fof(substitution_of_equivalents, axiom)
axiom_K fof(km5_axiom_K, conjecture)

LCL571+1.p Prove necessitation from the S1-0M10 axiomatization of S5
include('Axioms/LCL006+1.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+4.ax')
include('Axioms/LCL007+6.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
substitution_of_equivalents fof(substitution_of_equivalents, axiom)

necessitation fof(km5_necessitation, conjecture)

LCL572+1.p Prove axiom M from the S1-0M10 axiomatization of S5
include('Axioms/LCL006+1.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+4.ax')
include('Axioms/LCL007+6.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
substitution_of_equivalents fof(substitution_of_equivalents, axiom)
axiom_M fof(km5_axiom_M, conjecture)

LCL573+1.p Prove axiom 5 from the S1-0M10 axiomatization of S5
include('Axioms/LCL006+1.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+4.ax')
include('Axioms/LCL007+6.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
substitution_of_equivalents fof(substitution_of_equivalents, axiom)
$axiom_5$ fof(km5_$axiom_5$, conjecture)

LCL574+1.p Prove axiom 4 from the S1-0M10 axiomatization of S5
include('Axioms/LCL006+1.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+4.ax')
include('Axioms/LCL007+6.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
substitution_of_equivalents fof(substitution_of_equivalents, axiom)
$axiom_4$ fof(km4b_$axiom_4$, conjecture)

LCL575+1.p Prove axiom B from the S1-0M10 axiomatization of S5
include('Axioms/LCL006+1.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+4.ax')
include('Axioms/LCL007+6.ax')
op_or fof(hilbert_op_or, axiom)
op_implies_and fof(hilbert_op_implies_and, axiom)
op_equiv fof(hilbert_op_equiv, axiom)
substitution_of_equivalents fof(substitution_of_equivalents, axiom)
axiom_B fof(km4b_axiom_B, conjecture)

LCL576+1.p Prove axiom m6 from the S1-0M10 axiomatization of S5
include('Axioms/LCL006+1.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+4.ax')
include('Axioms/LCL007+6.ax')
axiom_m_6 fof(s1_0_m6s3m9b_axiom_m_6, conjecture)

LCL577+1.p Prove axiom s3 from the S1-0M10 axiomatization of S5
include('Axioms/LCL006+1.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+4.ax')

include('Axioms/LCL007+6.ax')
axiom_s$_3$ fof(s1_0_m6s3m9b_axiom_s$_3$, conjecture)

LCL578+1.p Prove axiom m9 from the S1-0M10 axiomatization of S5
include('Axioms/LCL006+1.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+4.ax')
include('Axioms/LCL007+6.ax')
axiom_m$_9$ fof(s1_0_m6s3m9b_axiom_m$_9$, conjecture)

LCL579^1.p Leibniz-equality definition means it's an equivalence
$\forall x$: \$i, y: \$i: ($\forall p$: \$i → \$o: ((p@x) ⇒ (p@y)) ⇒ $\forall p$: \$i → \$o: ((p@y) ⇒ (p@x))) thf(a, conjecture)

LCL579^2.p Leibniz-equality definition means it's an equivalence
cY: \$i thf(cY, type)
cX: \$i thf(cX, type)
$\forall p$: \$i → \$o: ((p@cY) ⇒ (p@cX)) ⇒ $\forall r$: \$i → \$o: ((r@cX) ⇒ (r@cY)) thf(cTHM76A, conjecture)

LCL580^1.p Popkorn problem 1
include('Axioms/LCL008^0.ax')
r: \$i → \$i → \$o thf(r, type)
mvalid@(mbox@r@mtrue) thf(thm, conjecture)

LCL581^1.p Popkorn problem 2
include('Axioms/LCL008^0.ax')
a: \$i → \$o thf(a, type)
r: \$i → \$i → \$o thf(r, type)
mvalid@(mimpl@(mbox@r@a)@(mbox@r@a)) thf(thm, conjecture)

LCL582^1.p Popkorn problem 3
include('Axioms/LCL008^0.ax')
a: \$i → \$o thf(a, type)
r: \$i → \$i → \$o thf(r, type)
s: \$i → \$i → \$o thf(s, type)
mvalid@(mimpl@(mbox@r@a)@(mbox@s@a)) thf(thm, conjecture)

LCL583^1.p Popkorn problem 4
include('Axioms/LCL008^0.ax')
a: \$i → \$o thf(a, type)
r: \$i → \$i → \$o thf(r, type)
s: \$i → \$i → \$o thf(s, type)
mvalid@(mbox@s@(mimpl@(mbox@r@a)@(mbox@r@a))) thf(thm, conjecture)

LCL584^1.p Popkorn problem 5
include('Axioms/LCL008^0.ax')
a: \$i → \$o thf(a, type)
b: \$i → \$o thf(b, type)
r: \$i → \$i → \$o thf(r, type)
s: \$i → \$i → \$o thf(s, type)
mvalid@(miff@(mbox@r@(mand@a@b))@(mand@(mbox@r@a)@(mbox@r@b))) thf(thm, conjecture)

LCL585^1.p Popkorn problem 6
include('Axioms/LCL008^0.ax')
a: \$i → \$o thf(a, type)
b: \$i → \$o thf(b, type)
r: \$i → \$i → \$o thf(r, type)
s: \$i → \$i → \$o thf(s, type)
mvalid@(mimpl@(mdia@r@(mimpl@a@b))@(mimpl@(mbox@r@a)@(mdia@r@b))) thf(thm, conjecture)

LCL586^1.p Popkorn problem 7
include('Axioms/LCL008^0.ax')
a: \$i → \$o thf(a, type)
b: \$i → \$o thf(b, type)
r: \$i → \$i → \$o thf(r, type)

mvalid@(mimpl@(mnot@(mdia@r@a))@(mbox@r@(mimpl@a@b)))   thf(thm, conjecture)

LCL587^1.p Popkorn problem 8
include('Axioms/LCL008^0.ax')
a: $i → $o   thf(a, type)
b: $i → $o   thf(b, type)
r: $i → $i → $o   thf(r, type)
mvalid@(mimpl@(mbox@r@b)@(mbox@r@(mimpl@a@b)))   thf(thm, conjecture)

LCL588^1.p Popkorn problem 9
include('Axioms/LCL008^0.ax')
a: $i → $o   thf(a, type)
b: $i → $o   thf(b, type)
r: $i → $i → $o   thf(r, type)
mvalid@(mimpl@(mimpl@(mdia@r@a)@(mbox@r@b))@(mbox@r@(mimpl@a@b)))   thf(thm, conjecture)

LCL589^1.p Popkorn problem 10
include('Axioms/LCL008^0.ax')
a: $i → $o   thf(a, type)
b: $i → $o   thf(b, type)
r: $i → $i → $o   thf(r, type)
mvalid@(mimpl@(mimpl@(mdia@r@a)@(mbox@r@b))@(mimpl@(mbox@r@a)@(mbox@r@b)))   thf(thm, conjecture)

LCL590^1.p Popkorn problem 11
include('Axioms/LCL008^0.ax')
a: $i → $o   thf(a, type)
b: $i → $o   thf(b, type)
r: $i → $i → $o   thf(r, type)
mvalid@(mimpl@(mimpl@(mdia@r@a)@(mbox@r@b))@(mimpl@(mdia@r@a)@(mdia@r@b)))   thf(thm, conjecture)

LCL591^1.p Axiom N is valid
include('Axioms/LCL008^0.ax')
∀r: $i → $i → $o, a: $i → $o: ((mvalid@a) ⇒ (mvalid@(mbox@r@a)))   thf(thm, conjecture)

LCL592^1.p Axiom D is valid
include('Axioms/LCL008^0.ax')
∀r: $i → $i → $o, a: $i → $o, b: $i → $o: (mvalid@(mimpl@(mbox@r@(mimpl@a@b))@(mimpl@(mbox@r@a)@(mbox@r@b)))

LCL593^1.p Is axiom T valid in K?
include('Axioms/LCL008^0.ax')
∀r: $i → $i → $o, a: $i → $o: (mvalid@(mimpl@(mbox@r@a)@a))   thf(thm, conjecture)

LCL594^1.p Relation for all propositions making T valid in K
Is there a relation R such that for all modal propositions A, axiom T is valid in K
include('Axioms/LCL008^0.ax')
∃r: $i → $i → $o: ∀a: $i → $o: (mvalid@(mimpl@(mbox@r@a)@a))   thf(thm, conjecture)

LCL595^1.p Is axiom T equivalent to reflexivity of R in K
include('Axioms/LCL008^0.ax')
include('Axioms/SET008^2.ax')
r: $i → $i → $o   thf(r, type)
∀a: $i → $o: (mvalid@(mimpl@(mbox@r@a)@a)) ⟺ (reflexive@r)   thf(thm, conjecture)

LCL596^1.p Is axiom 4 valid in K?
include('Axioms/LCL008^0.ax')
∀r: $i → $i → $o, a: $i → $o: (mvalid@(mimpl@(mbox@r@a)@(mbox@r@(mbox@r@a))))   thf(thm, conjecture)

LCL597^1.p Relation and proposition for 4 in K
Is there a relation R and a modal proposition A for which axiom 4 is valid in K?
include('Axioms/LCL008^0.ax')
∃r: $i → $i → $o: ∀a: $i → $o: (mvalid@(mimpl@(mbox@r@a)@(mbox@r@(mbox@r@a))))   thf(thm, conjecture)

LCL598^1.p Is axiom 4 equivalent to irreflexivity?
include('Axioms/LCL008^0.ax')
include('Axioms/SET008^2.ax')
∀r: $i → $i → $o: (∀a: $i → $o: (mvalid@(mimpl@(mbox@r@a)@(mbox@r@(mbox@r@a)))) ⟺ (irreflexive@r))   thf(th

LCL599∧1.p Is axiom 4 equivalent to symmetry?
include('Axioms/LCL008^0.ax')
include('Axioms/SET008^2.ax')
$\forall r$: \$i → \$i → \$o: ($\forall a$: \$i → \$o: (mvalid@(mimpl@(mbox@r@a)@(mbox@r@(mbox@r@a)))) ⟺ (symmetric@r)) thf(th

LCL600∧1.p Is axiom 4 equivalent to transitivity of R in K?
include('Axioms/LCL008^0.ax')
include('Axioms/SET008^2.ax')
$\forall r$: \$i → \$i → \$o: ($\forall a$: \$i → \$o: (mvalid@(mimpl@(mbox@r@a)@(mbox@r@(mbox@r@a)))) ⟺ (transitive@r)) thf(thr

LCL601∧1.p Axiom 4 for all R means all R are valid
If axiom 4 is valid for all relations R then all relations R are transitive.
include('Axioms/LCL008^0.ax')
include('Axioms/SET008^2.ax')
$\forall r$: \$i → \$i → \$o, a: \$i → \$o: (mvalid@(mimpl@(mbox@r@a)@(mbox@r@(mbox@r@a)))) ⟺ $\forall r$: \$i → \$i → \$o: (transitive@r) thf(thm, conjecture)

LCL602∧1.p T and 4 equivalent to reflexivity and transitivity of R in K
include('Axioms/LCL008^0.ax')
include('Axioms/SET008^2.ax')
$\forall r$: \$i → \$i → \$o: ($\forall a$: \$i → \$o: (mvalid@(mimpl@(mbox@r@a)@(mbox@r@(mbox@r@a))) and mvalid@(mimpl@(mbox@r@
(reflexive@r and transitive@r)) thf(thm, conjecture)

LCL603∧1.p T and 4 imply reflexivity and transitivity of R in K
include('Axioms/LCL008^0.ax')
include('Axioms/SET008^2.ax')
$\forall r$: \$i → \$i → \$o: ($\forall a$: \$i → \$o: (mvalid@(mimpl@(mbox@r@a)@(mbox@r@(mbox@r@a))) and mvalid@(mimpl@(mbox@r@
(reflexive@r and transitive@r)) thf(thm, conjecture)

LCL604∧1.p T and 4 implied by reflexivity and transitivity of R in K
include('Axioms/LCL008^0.ax')
include('Axioms/SET008^2.ax')
$\forall r$: \$i → \$i → \$o: ($\forall a$: \$i → \$o: (mvalid@(mimpl@(mbox@r@a)@(mbox@r@(mbox@r@a))) and mvalid@(mimpl@(mbox@r@
(reflexive@r and transitive@r)) thf(thm, conjecture)

LCL606∧1.p LAMBDA∧mm_1 validates the Barcan formula axioms
include('Axioms/LCL008^0.ax')
$\forall r$: \$i → \$i → \$o, a: individuals → \$i → \$o: (mvalid@(mimpl@(mall@λx: individuals: (mbox@r@(a@x)))@(mbox@r@(mall@λ

LCL607∧1.p LAMBDA∧mm_1 validates the axioms defining possibility
include('Axioms/LCL008^0.ax')
$\forall r$: \$i → \$i → \$o, a: \$i → \$o: (mvalid@(miff@(mdia@r@a)@(mnot@(mbox@r@(mnot@a))))) thf(thm, conjecture)

LCL608∧1.p LAMBDA∧mm_1 validates the modus ponens rule
include('Axioms/LCL008^0.ax')
$\forall a$: \$i → \$o, b: \$i → \$o: ((mvalid@a and mvalid@(mimpl@a@b)) ⇒ (mvalid@b)) thf(thm, conjecture)

LCL609∧1.p LAMBDA∧mm_1 validates the generalization rule
include('Axioms/LCL008^0.ax')
$\forall p$: individuals → \$i → \$o: ($\forall x$: individuals: (mvalid@(p@x)) ⇒ (mvalid@(mall@λx: individuals: (p@x)))) thf(thm, conje

LCL611∧1.p LAMBDA∧mm_1 validates the converse Barcan formula
include('Axioms/LCL008^0.ax')
$\forall r$: \$i → \$i → \$o, a: individuals → \$i → \$o: (mvalid@(mimpl@(mbox@r@(mall@λx: individuals: (a@x)))@(mall@λx: individu

LCL612∧1.p Modus Ponens holds in K
include('Axioms/LCL008^0.ax')
$\forall r$: \$i → \$i → \$o, x: \$i → \$o, y: \$i → \$o: ((mvalid@x and mvalid@(mimpl@x@y)) ⇒ (mvalid@y)) thf(modus_ponens, con

LCL613∧1.p Simple theorem of K
include('Axioms/LCL008^0.ax')
$\forall r$: \$i → \$i → \$o, x: \$i → \$o, y: \$i → \$o: (mvalid@(mimpl@(mbox@r@(mand@x@y))@(mbox@r@x))) thf(thm, conjecture

LCL614∧1.p Regularity is a derived rule in K
include('Axioms/LCL008^0.ax')
$\forall r$: \$i → \$i → \$o, x: \$i → \$o, y: \$i → \$o: ((mvalid@(mimpl@x@y)) ⇒ (mvalid@(mimpl@(mbox@r@x)@(mbox@r@x)))) t

LCL615∧1.p Axiom KB

include('Axioms/LCL008^0.ax')
$\forall r$: \$i → \$i → \$o, x: \$i → \$o: (mvalid@(mimpl@x@(mbox@r@(mdia@r@x)))) thf(kb, conjecture)

LCL617∧1.p Axiom GL - the Loeb formula
include('Axioms/LCL008^0.ax')
$\forall r$: \$i → \$i → \$o, x: \$i → \$o: (mvalid@(mimpl@(mbox@r@(mimpl@(mbox@r@x)@x))@(mbox@r@x))) thf(thm, conjectu

LCL618∧1.p Axiom GL implies Axiom K4 in K
include('Axioms/LCL008^0.ax')
$\forall r$: \$i → \$i → \$o, x: \$i → \$o: (mvalid@(mimpl@(mbox@r@(mimpl@(mbox@r@x)@x))@(mbox@r@x))) thf(gl, axiom)
$\forall r$: \$i → \$i → \$o, x: \$i → \$o: (mvalid@(mimpl@(mbox@r@x)@(mbox@r@(mbox@r@(mbox@r@x))))) thf(k_4, conjecture)

LCL619∧1.p A simple theorem of K4
include('Axioms/LCL008^0.ax')
$\forall r$: \$i → \$i → \$o, x: \$i → \$o: (mvalid@(mimpl@(mbox@r@x)@(mbox@r@(mbox@r@x)))) thf(k_4, axiom)
$\forall r$: \$i → \$i → \$o, x: \$i → \$o, y: \$i → \$o: (mvalid@(mimpl@(mbox@r@(mimpl@x@y))@(mimpl@(mbox@r@x)@(mbox@r@(m

LCL620∧1.p A simple theorem of propositional logic
include('Axioms/LCL008^0.ax')
$\forall r$: \$i → \$i → \$o, x: \$i → \$o, y: \$i → \$o: (mvalid@(mimpl@(mnot@(mand@x@y))@(mor@(mnot@x)@(mnot@y)))) thf(th

LCL621∧1.p A simple theorem of K4
include('Axioms/LCL008^0.ax')
$\forall r$: \$i → \$i → \$o, x: \$i → \$o: (mvalid@(mimpl@(mbox@r@x)@(mbox@r@(mbox@r@x)))) thf(k_4, axiom)
$\forall r$: \$i → \$i → \$o, x: \$i → \$o, y: \$i → \$o: (mvalid@(mimpl@(mand@(mbox@r@(mdia@r@x))@(mbox@r@y))@(mbox@r@(md

LCL623∧1.p The Loeb formula is a theorem in GL
include('Axioms/LCL008^0.ax')
$\forall r$: \$i → \$i → \$o, x: \$i → \$o: (mvalid@(mimpl@(mbox@r@(mimpl@(mbox@r@x)@x))@(mbox@r@x))) thf(gl, axiom)
$\forall r$: \$i → \$i → \$o, p: \$i → \$o: (mvalid@(mimpl@(mbox@r@(mimpl@(mbox@r@p)@p))@(mbox@r@p))) thf(loeb, conjectur

LCL624∧1.p A simple theorem of K
include('Axioms/LCL008^0.ax')
$\forall r$: \$i → \$i → \$o, p: \$i → \$o, q: \$i → \$o: (mvalid@(mimpl@(mand@(mdia@r@(mbox@r@p))@(mbox@r@(mdia@r@q)))@(md

LCL625∧1.p GL/K4 axiom is valid in this frame
In a frame that is transitive and upwards well-founded, the GL/K4 axiom is valid.
include('Axioms/LCL008^0.ax')
include('Axioms/SET008^2.ax')
$\forall r$: \$i → \$i → \$o: (transitive@r and upwards_well_founded@r) thf(upwf_trans, axiom)
$\forall r$: \$i → \$i → \$o, p: \$i → \$o: (mvalid@(mimpl@(mbox@r@(mimpl@(mbox@r@p)@p))@(mbox@r@p))) thf(loeb, conjectur

LCL626∧1.p Loeb axiom is valid in this frame
In a frame that is transitive and upwards well-founded, the Loeb axiom is valid.
include('Axioms/LCL008^0.ax')
include('Axioms/SET008^2.ax')
$\forall r$: \$i → \$i → \$o: (transitive@r and upwards_well_founded@r) thf(upwf_trans, axiom)
$\forall r$: \$i → \$i → \$o, x: \$i → \$o: (mvalid@(mimpl@(mbox@r@x)@(mbox@r@(mbox@r@(mbox@r@x))))) thf(k_4, conjecture)

LCL629∧1.p Simple theorem about knowledge
include('Axioms/LCL008^0.ax')
a: \$i → \$i → \$o thf(a, type)
b: \$i → \$i → \$o thf(b, type)
c: \$i → \$i → \$o thf(c, type)
$\forall x$: \$i → \$o, r: \$i → \$i → \$o: (mvalid@(mimpl@(mbox@r@x)@x)) thf(knowledge_implies_truth, axiom)
$\forall x$: \$i → \$o, r: \$i → \$i → \$o: (mvalid@(mimpl@(mbox@r@x)@(mbox@r@(mbox@r@x)))) thf(positive_introspection, axio
$\forall x$: \$i → \$o, r: \$i → \$i → \$o: (mvalid@(mimpl@(mnot@(mbox@r@x))@(mbox@r@(mnot@(mbox@r@x))))) thf(negitive_i
$\forall x$: \$i → \$o, y: \$i → \$o: (mvalid@(mimpl@(mand@(mbox@a@(mnot@(mbox@b@(mnot@(mbox@b@(mnot@y))))))@(mbox@a

LCL630∧1.p The muddy forehead puzzle
include('Axioms/LCL008^0.ax')
a: \$i → \$i → \$o thf(a, type)
b: \$i → \$i → \$o thf(b, type)
c: \$i → \$i → \$o thf(c, type)
mfa: \$i → \$o thf(mfa, type)
mfb: \$i → \$o thf(mfb, type)

mfc: $i → $o thf(mfc, type)
ck: ($i → $o) → $i → $o thf(ck, type)
s: $i → $o thf(*s*, type)
∀*x*: $i → $o, *r*: $i → $i → $o: (mvalid@(mimpl@(mbox@*r*@*x*)@*x*)) thf(knowledge_implies_truth, axiom)
∀*x*: $i → $o, *r*: $i → $i → $o: (mvalid@(mimpl@(mbox@*r*@*x*)@(mbox@*r*@(mbox@*r*@*x*)))) thf(positive_introspection, axio
∀*x*: $i → $o, *r*: $i → $i → $o: (mvalid@(mimpl@(mnot@(mbox@*r*@*x*))@(mbox@*r*@(mnot@(mbox@*r*@*x*))))) thf(negitive_i
ck = (λ*x*: $i → $o, *w*: $i: ∀*r*: $i → $i → $o: (mbox@*r*@*x*@*w*)) thf(common_knowledge, definition)
mvalid@(ck@(mor@(mbox@*a*@mfb)@(mbox@*a*@(mnot@mfb)))) thf(what_a_knows_about_b, axiom)
mvalid@(ck@(mor@(mbox@*a*@mfc)@(mbox@*a*@(mnot@mfc)))) thf(what_a_knows_about_c, axiom)
mvalid@(ck@(mor@(mbox@*b*@mfa)@(mbox@*b*@(mnot@mfa)))) thf(what_b_knows_about_a, axiom)
mvalid@(ck@(mor@(mbox@*b*@mfc)@(mbox@*b*@(mnot@mfc)))) thf(what_b_knows_about_c, axiom)
mvalid@(ck@(mor@(mbox@*c*@mfa)@(mbox@*c*@(mnot@mfa)))) thf(what_c_knows_about_a, axiom)
mvalid@(ck@(mor@(mbox@*c*@mfb)@(mbox@*c*@(mnot@mfb)))) thf(what_c_knows_about_b, axiom)
s = (mor@(mbox@*a*@mfa)@(mor@(mbox@*a*@(mnot@mfa))@(mor@(mbox@*b*@mfb)@(mor@(mbox@*b*@(mnot@mfb))@(mor@
mvalid@(mnot@(mimpl@(ck@(mor@mfa@(mor@mfb@mfc)))@*s*)) thf(thm, conjecture)

LCL631∧1.p The muddy forehead puzzle
include('Axioms/LCL008^0.ax')
a: $i → $i → $o thf(*a*, type)
b: $i → $i → $o thf(*b*, type)
c: $i → $i → $o thf(*c*, type)
mfa: $i → $o thf(mfa, type)
mfb: $i → $o thf(mfb, type)
mfc: $i → $o thf(mfc, type)
ck: ($i → $o) → $i → $o thf(ck, type)
s: $i → $o thf(*s*, type)
∀*x*: $i → $o, *r*: $i → $i → $o: (mvalid@(mimpl@(mbox@*r*@*x*)@*x*)) thf(knowledge_implies_truth, axiom)
∀*x*: $i → $o, *r*: $i → $i → $o: (mvalid@(mimpl@(mbox@*r*@*x*)@(mbox@*r*@(mbox@*r*@*x*)))) thf(positive_introspection, axio
∀*x*: $i → $o, *r*: $i → $i → $o: (mvalid@(mimpl@(mnot@(mbox@*r*@*x*))@(mbox@*r*@(mnot@(mbox@*r*@*x*))))) thf(negitive_i
ck = (λ*x*: $i → $o, *w*: $i: ∀*r*: $i → $i → $o: (mbox@*r*@*x*@*w*)) thf(common_knowledge, definition)
mvalid@(ck@(mor@(mbox@*a*@mfb)@(mbox@*a*@(mnot@mfb)))) thf(what_a_knows_about_b, axiom)
mvalid@(ck@(mor@(mbox@*a*@mfc)@(mbox@*a*@(mnot@mfc)))) thf(what_a_knows_about_c, axiom)
mvalid@(ck@(mor@(mbox@*b*@mfa)@(mbox@*b*@(mnot@mfa)))) thf(what_b_knows_about_a, axiom)
mvalid@(ck@(mor@(mbox@*b*@mfc)@(mbox@*b*@(mnot@mfc)))) thf(what_b_knows_about_c, axiom)
mvalid@(ck@(mor@(mbox@*c*@mfa)@(mbox@*c*@(mnot@mfa)))) thf(what_c_knows_about_a, axiom)
mvalid@(ck@(mor@(mbox@*c*@mfb)@(mbox@*c*@(mnot@mfb)))) thf(what_c_knows_about_b, axiom)
s = (mor@(mbox@*a*@mfa)@(mor@(mbox@*a*@(mnot@mfa))@(mor@(mbox@*b*@mfb)@(mor@(mbox@*b*@(mnot@mfb))@(mor@
mvalid@(mnot@(mimpl@(ck@(mnot@(mimpl@(ck@(mor@mfa@(mor@mfb@mfc)))@*s*)))@*s*)) thf(thm, conjecture)

LCL632∧1.p The muddy forehead puzzle
include('Axioms/LCL008^0.ax')
a: $i → $i → $o thf(*a*, type)
b: $i → $i → $o thf(*b*, type)
c: $i → $i → $o thf(*c*, type)
mfa: $i → $o thf(mfa, type)
mfb: $i → $o thf(mfb, type)
mfc: $i → $o thf(mfc, type)
ck: ($i → $o) → $i → $o thf(ck, type)
s: $i → $o thf(*s*, type)
∀*x*: $i → $o, *r*: $i → $i → $o: (mvalid@(mimpl@(mbox@*r*@*x*)@*x*)) thf(knowledge_implies_truth, axiom)
∀*x*: $i → $o, *r*: $i → $i → $o: (mvalid@(mimpl@(mbox@*r*@*x*)@(mbox@*r*@(mbox@*r*@*x*)))) thf(positive_introspection, axio
∀*x*: $i → $o, *r*: $i → $i → $o: (mvalid@(mimpl@(mnot@(mbox@*r*@*x*))@(mbox@*r*@(mnot@(mbox@*r*@*x*))))) thf(negitive_i
ck = (λ*x*: $i → $o, *w*: $i: ∀*r*: $i → $i → $o: (mbox@*r*@*x*@*w*)) thf(common_knowledge, definition)
mvalid@(ck@(mor@(mbox@*a*@mfb)@(mbox@*a*@(mnot@mfb)))) thf(what_a_knows_about_b, axiom)
mvalid@(ck@(mor@(mbox@*a*@mfc)@(mbox@*a*@(mnot@mfc)))) thf(what_a_knows_about_c, axiom)
mvalid@(ck@(mor@(mbox@*b*@mfa)@(mbox@*b*@(mnot@mfa)))) thf(what_b_knows_about_a, axiom)
mvalid@(ck@(mor@(mbox@*b*@mfc)@(mbox@*b*@(mnot@mfc)))) thf(what_b_knows_about_c, axiom)
mvalid@(ck@(mor@(mbox@*c*@mfa)@(mbox@*c*@(mnot@mfa)))) thf(what_c_knows_about_a, axiom)
mvalid@(ck@(mor@(mbox@*c*@mfb)@(mbox@*c*@(mnot@mfb)))) thf(what_c_knows_about_b, axiom)
s = (mor@(mbox@*a*@mfa)@(mor@(mbox@*a*@(mnot@mfa))@(mor@(mbox@*b*@mfb)@(mor@(mbox@*b*@(mnot@mfb))@(mor@

mvalid@(mimpl@(ck@(mnot@(mimpl@(ck@(mnot@(mimpl@(ck@(mor@mfa@(mor@mfb@mfc)))@s)))@s)))@s) thf(thm,

LCL633^1.p Goedel's ontological argument on the existence of God
include('Axioms/LCL008^0.ax')
a: \$tType thf(a_type, type)
p: (a → \$i → \$o) → \$i → \$o thf(p_type, type)
g: a → \$i → \$o thf(g_type, type)
e: (a → \$i → \$o) → a → \$i → \$o thf(e_type, type)
r: \$i → \$i → \$o thf(r_type, type)
∀x: a → \$i → \$o: (mvalid@(mimpl@(mnot@(p@x))@(p@λz: a: (mnot@(x@z))))) thf(positiveness, axiom)
g = (λz: a, w: \$i: ∀x: a → \$i → \$o: (mimpl@(p@x)@(x@z)@w)) thf(g, definition)
e = (λx: a → \$i → \$o, z: a, p: \$i: ∀y: a → \$i → \$o: (mimpl@(y@z)@(mbox@r@λq: \$i: ∀w: a: (mimpl@(x@w)@(y@w)@q))@p))
mvalid@λw: \$i: ∀z: a: (mimpl@(g@z)@(e@g@z)@w) thf(thm, conjecture)

LCL636+1.001.p In K, the branching formula made provable, size 1
The branching formula plus a negation symbol in front and an additional subformula to make the formula provable.
¬∃x: ¬¬∀y: (¬$r_1(x, y)$ or $p_2(y)$) or ¬∀y: (¬$r_1(x, y)$ or (((¬∀x: (¬$r_1(y, x)$ or ¬¬$p_2(x)$ and ¬$p_{102}(x)$ and $p_{101}(x)$) and ¬∀x

LCL637+1.001.p In K, the branching formula, size 1
¬∃x: (∀y: (¬$r_1(x, y)$ or (((¬∀x: (¬$r_1(y, x)$ or ¬¬$p_2(x)$ and ¬$p_{102}(x)$ and $p_{101}(x)$) and ¬∀x: (¬$r_1(y, x)$ or ¬$p_2(x)$ and ¬p_{10}

LCL638+1.001.p In K, D & A4 & B p0/p0 → T, size 1
¬∃x: ¬¬∀y: (¬$r_1(x, y)$ or ¬¬∀x: (¬$r_1(y, x)$ or ¬∀y: (¬$r_1(x, y)$ or $p_1(y)$)) and ¬$p_1(y)$) or ¬∀y: (¬$r_1(x, y)$ or ¬¬∀x: (¬$r_1($

LCL639+1.001.p In K, A5 not provable with instances of D, A4, and T, size 1
¬∃x: ¬¬∀y: (¬$r_1(x, y)$ or ¬¬∀x: (¬$r_1(y, x)$ or ∀y: (¬$r_1(x, y)$ or ¬¬$p_1(y)$ and ∀x: (¬$r_1(y, x)$ or $p_1(x)$)))) and ∀x: (¬$r_1(y, x)$

LCL644+1.001.p In K, H2 → L, size 1
¬∃x: ¬¬∀y: (¬$r_1(x, y)$ or ¬$p_2(y)$ and ∀x: (¬$r_1(y, x)$ or $p_2(x)$) or $p_1(y)$) or ∀y: (¬$r_1(x, y)$ or $p_2(y)$ or ¬$p_1(y)$ and ∀x: (¬$r_1($

LCL645+1.001.p In K, L+ is not provable with instances of L, size 1
¬∃x: ¬∀y: (¬$r_1(x, y)$ or $p_1(y)$ or ¬∀x: (¬$r_1(y, x)$ or $p_1(x)$)) or ∀y: (¬$r_1(x, y)$ or $p_1(y)$ or ¬∀x: (¬$r_1(y, x)$ or $p_1(x)$)) fof(

LCL646+1.001.p In K, path through a labyrinth, size 1
¬∃x: ¬¬∀y: (¬$r_1(x, y)$ or $p_6(y)$) or ¬∀y: (¬$r_1(x, y)$ or $p_2(y)$) or ¬∀y: (¬$r_1(x, y)$ or $p_4(y)$) or ¬∀y: (¬$r_1(x, y)$ or $p_2(y)$) or ∀

LCL648+1.001.p In K, pigeonhole formulae, size 1
¬∃x: ¬¬∀y: (¬$r_1(x, y)$ or ¬$p_{201}(y)$ and $p_{101}(y)$) or ∀y: (¬$r_1(x, y)$ or ¬$p_{201}(y)$ and $p_{101}(y)$) fof(main, conjecture)

LCL649+1.001.p In K, pigeonhole formulae missing a conjunct, size 1
¬∃x: ¬¬∀y: (¬$r_1(x, y)$ or ¬$p_{201}(y)$ and ¬$p_{101}(y)$) or ∀y: (¬$r_1(x, y)$ or ¬$p_{201}(y)$ and $p_{101}(y)$) fof(main, conjecture)

LCL650+1.001.p In K, black and white polygon with odd number of vertices, size 1
If we have a polygon with n vertices, and all the vertices are either black or white, then two adjacent vertices have the same colour.
¬∃x: ¬∀y: (¬$r_1(x, y)$ or ∀x: (¬$r_1(y, x)$ or ∀y: (¬$r_1(x, y)$ or ∀x: (¬$r_1(y, x)$ or (¬$p_8(x)$ and ¬$p_6(x)$ and ¬$p_4(x)$ and ¬$p_2(x)$))

LCL654+1.001.p In KT, A5box p0/p0 & box A5 p0/p0 → A4, size 1
∀x: $r_1(x, x)$ fof(reflexivity, axiom)
¬∃x: ¬¬∀y: (¬$r_1(x, y)$ or ¬¬∀x: (¬$r_1(y, x)$ or ¬∀y: (¬$r_1(x, y)$ or ¬$p_1(y)$))) and $p_1(y)$) or ¬∀y: (¬$r_1(x, y)$ or ∀x: (¬$r_1(y, x$

LCL655+1.001.p In KT, A5 not provable with instances of A4, size 1
∀x: $r_1(x, x)$ fof(reflexivity, axiom)
¬∃x: ¬¬∀y: (¬$r_1(x, y)$ or ¬¬∀x: (¬$r_1(y, x)$ or ∀y: (¬$r_1(x, y)$ or ¬¬$p_1(y)$ and ∀x: (¬$r_1(y, x)$ or $p_1(x)$)))) and ∀x: (¬$r_1(y, x)$

LCL656+1.001.p In KT, the branching formula made provable, size 1
The branching formula plus a negation symbol in front and an additional subformula to make the formula provable.
∀x: $r_1(x, x)$ fof(reflexivity, axiom)
¬∃x: ¬¬∀y: (¬$r_1(x, y)$ or $p_2(y)$) or ¬∀y: (¬$r_1(x, y)$ or (((¬∀x: (¬$r_1(y, x)$ or ¬¬$p_2(x)$ and ¬$p_{102}(x)$ and $p_{101}(x)$) and ¬∀x

LCL657+1.001.p In KT, the branching formula, size 1
∀x: $r_1(x, x)$ fof(reflexivity, axiom)
¬∃x: (∀y: (¬$r_1(x, y)$ or (((¬∀x: (¬$r_1(y, x)$ or ¬¬$p_2(x)$ and ¬$p_{102}(x)$ and $p_{101}(x)$) and ¬∀x: (¬$r_1(y, x)$ or ¬$p_2(x)$ and ¬p_{10}

LCL662+1.001.p In KT, in backward search find a way through box and dia, size 1
¬∃x: ¬¬$p_1(x)$ or $p_1(x)$ fof(main, conjecture)

LCL663+1.001.p In KT, in backwards search no way through box and dia, size 1
¬∃x: ¬$p_1(x)$ fof(main, conjecture)

LCL664+1.001.p In KT, path through a labyrinth, size 1

$\forall x\colon r_1(x,x)$ fof(reflexivity, axiom)
$\neg\,\exists x\colon \neg\,\neg\,\forall y\colon (\neg\, r_1(x,y) \text{ or } p_{16}(y)) \text{ or } \neg\,\forall y\colon (\neg\, r_1(x,y) \text{ or } p_{12}(y)) \text{ or } \neg\,\forall y\colon (\neg\, r_1(x,y) \text{ or } p_{14}(y)) \text{ or } \neg\,\forall y\colon (\neg\, r_1(x,y) \text{ or } p_{12}(y))$

LCL666+1.001.p In KT, pigeonhole formulae, size 1
$\forall x\colon r_1(x,x)$ fof(reflexivity, axiom)
$\neg\,\exists x\colon \neg\,\neg\,\forall y\colon (\neg\, r_1(x,y) \text{ or } \neg\, p_{201}(y) \text{ and } p_{101}(y)) \text{ or } \neg\, p_{201}(x) \text{ and } p_{101}(x)$ fof(main, conjecture)

LCL667+1.001.p In KT, pigeonhole formulae missing a conjunct, size 1
$\forall x\colon r_1(x,x)$ fof(reflexivity, axiom)
$\neg\,\exists x\colon \neg\,\neg\,\forall y\colon (\neg\, r_1(x,y) \text{ or } \neg\, p_{201}(y) \text{ and } \neg\, p_{101}(y)) \text{ or } \neg\, p_{201}(x) \text{ and } p_{101}(x)$ fof(main, conjecture)

LCL672+1.001.p In S4, A5box p0/p0 & box A5 p0/p0 → A5, size 1
$\forall x\colon r_1(x,x)$ fof(reflexivity, axiom)
$\forall x,y,z\colon ((r_1(x,y) \text{ and } r_1(y,z)) \Rightarrow r_1(x,z))$ fof(transitivity, axiom)
$\neg\,\exists x\colon \neg\,(\neg\,\forall y\colon (\neg\, r_1(x,y) \text{ or } p_2(y)) \text{ and } \neg\,\forall y\colon (\neg\, r_1(x,y) \text{ or } \neg\,\neg\,\forall x\colon (\neg\, r_1(y,x) \text{ or } \neg\,\forall y\colon (\neg\, r_1(x,y) \text{ or } \neg\, p_1(y))) \text{ and } p_1(y)) \text{ and}$

LCL674+1.001.p In S4, the branching formula made provable, size 1
The branching formula plus a negation symbol in front and an additional subformula to make the formula provable.
$\forall x\colon r_1(x,x)$ fof(reflexivity, axiom)
$\forall x,y,z\colon ((r_1(x,y) \text{ and } r_1(y,z)) \Rightarrow r_1(x,z))$ fof(transitivity, axiom)
$\neg\,\exists x\colon \neg\,\neg\,\forall y\colon (\neg\, r_1(x,y) \text{ or } p_2(y)) \text{ or } \neg\,\forall y\colon (\neg\, r_1(x,y) \text{ or } (((\neg\,\forall x\colon (\neg\, r_1(y,x) \text{ or } \neg\,\neg\, p_2(x) \text{ and } \neg\, p_{102}(x) \text{ and } p_{101}(x)) \text{ and } \neg\,\forall x$

LCL675+1.001.p In S4, the branching formula, size 1
$\forall x\colon r_1(x,x)$ fof(reflexivity, axiom)
$\forall x,y,z\colon ((r_1(x,y) \text{ and } r_1(y,z)) \Rightarrow r_1(x,z))$ fof(transitivity, axiom)
$\neg\,\exists x\colon (\forall y\colon (\neg\, r_1(x,y) \text{ or } (((\neg\,\forall x\colon (\neg\, r_1(y,x) \text{ or } \neg\,\neg\, p_2(x) \text{ and } \neg\, p_{102}(x) \text{ and } p_{101}(x)) \text{ and } \neg\,\forall x\colon (\neg\, r_1(y,x) \text{ or } \neg\, p_2(x) \text{ and } \neg\, p_{10}$

LCL678+1.001.p In S4, formula provable in intuitionistic logic, size 1
$\forall x\colon r_1(x,x)$ fof(reflexivity, axiom)
$\forall x,y,z\colon ((r_1(x,y) \text{ and } r_1(y,z)) \Rightarrow r_1(x,z))$ fof(transitivity, axiom)
$\neg\,\exists x\colon \neg\,\$\text{false} \text{ or } \neg\,\forall y\colon (\neg\, r_1(x,y) \text{ or } \$\text{false} \text{ or } \neg\,\forall x\colon (\neg\, r_1(y,x) \text{ or } \forall y\colon (\neg\, r_1(x,y) \text{ or } p_1(y)) \text{ or } \neg\,\forall y\colon (\neg\, r_1(x,y) \text{ or } p_1(y))))$ f

LCL679+1.001.p In S4, formula not provable in intuitionistic logic, size 1
$\neg\,\exists x\colon \neg\,\$\text{false} \text{ or } \$\text{false}$ fof(main, conjecture)

LCL680+1.001.p In S4, in backward search find a way through box and dia, size 1
$\neg\,\exists x\colon \neg\,\neg\, p_1(x) \text{ or } p_1(x)$ fof(main, conjecture)

LCL681+1.001.p In S4, in backwards search no way through box and dia, size 1
$\neg\,\exists x\colon \neg\, p_1(x)$ fof(main, conjecture)

LCL682+1.001.p In S4, path through a labyrinth, size 1
$\forall x\colon r_1(x,x)$ fof(reflexivity, axiom)
$\forall x,y,z\colon ((r_1(x,y) \text{ and } r_1(y,z)) \Rightarrow r_1(x,z))$ fof(transitivity, axiom)
$\neg\,\exists x\colon \neg\,\neg\,\forall y\colon (\neg\, r_1(x,y) \text{ or } \neg\,\neg\,\forall x\colon (\neg\, r_1(y,x) \text{ or } p_{16}(x)) \text{ or } \neg\,\forall x\colon (\neg\, r_1(y,x) \text{ or } p_{12}(x)) \text{ or } \neg\,\forall x\colon (\neg\, r_1(y,x) \text{ or } p_{14}(x)) \text{ or } \neg\,\forall x$

LCL684+1.001.p In S4, pigeonhole formulae, size 1
$\forall x\colon r_1(x,x)$ fof(reflexivity, axiom)
$\forall x,y,z\colon ((r_1(x,y) \text{ and } r_1(y,z)) \Rightarrow r_1(x,z))$ fof(transitivity, axiom)
$\neg\,\exists x\colon \neg\,\neg\,\forall y\colon (\neg\, r_1(x,y) \text{ or } \forall x\colon (\neg\, r_1(y,x) \text{ or } \neg\, p_{201}(x) \text{ and } p_{101}(x))) \text{ or } \neg\, p_{201}(x) \text{ and } p_{101}(x)$ fof(main, conjecture)

LCL685+1.001.p In S4, pigeonhole formulae missing a conjunct, size 1
$\forall x\colon r_1(x,x)$ fof(reflexivity, axiom)
$\forall x,y,z\colon ((r_1(x,y) \text{ and } r_1(y,z)) \Rightarrow r_1(x,z))$ fof(transitivity, axiom)
$\neg\,\exists x\colon \neg\,\neg\,\forall y\colon (\neg\, r_1(x,y) \text{ or } \forall x\colon (\neg\, r_1(y,x) \text{ or } \neg\, p_{201}(x) \text{ and } \neg\, p_{101}(x))) \text{ or } \neg\, p_{201}(x) \text{ and } p_{101}(x)$ fof(main, conjecture)

LCL686+1.001.p In S4, formula provable in S5 embedding, size 1
$\forall x\colon r_1(x,x)$ fof(reflexivity, axiom)
$\forall x,y,z\colon ((r_1(x,y) \text{ and } r_1(y,z)) \Rightarrow r_1(x,z))$ fof(transitivity, axiom)
$\neg\,\exists x\colon \neg\,\forall y\colon (\neg\, r_1(x,y) \text{ or } \neg\, p_3(y) \text{ or } \forall x\colon (\neg\, r_1(y,x) \text{ or } \neg\, p_1(x))) \text{ or } \forall y\colon (\neg\, r_1(x,y) \text{ or } \neg\,\forall x\colon (\neg\, r_1(y,x) \text{ or } \neg\,\forall y\colon (\neg\, r_1(x,y) \text{ or } \$\text{f}$

LCL687+1.001.p In S4, formula not provable in S5 embedding, size 1
$\forall x\colon r_1(x,x)$ fof(reflexivity, axiom)
$\forall x,y,z\colon ((r_1(x,y) \text{ and } r_1(y,z)) \Rightarrow r_1(x,z))$ fof(transitivity, axiom)
$\neg\,\exists x\colon \neg\,\forall y\colon (\neg\, r_1(x,y) \text{ or } \neg\, p_6(y) \text{ or } \forall x\colon (\neg\, r_1(y,x) \text{ or } \neg\, p_1(x))) \text{ or } \forall y\colon (\neg\, r_1(x,y) \text{ or } \neg\,\forall x\colon (\neg\, r_1(y,x) \text{ or } \neg\,\forall y\colon (\neg\, r_1(x,y) \text{ or } p_6$

LCL688+1.001.p In S4, formula with T and A4, size 1
Tdia p0/p0 & box T box dia p0/p0 & A4dia p0/p0 & box(dia box dia p0 → (p0 → box p0)) → dia box p0 — dia box p0.

$\forall x\colon r_1(x,x)$ fof(reflexivity, axiom)
$\forall x,y,z\colon ((r_1(x,y) \text{ and } r_1(y,z)) \Rightarrow r_1(x,z))$ fof(transitivity, axiom)
$\neg\exists x\colon \neg\neg\forall y\colon (\neg r_1(x,y) \text{ or } \neg p_4(y)) \text{ or } \neg\forall y\colon (\neg r_1(x,y) \text{ or } \neg\neg\forall x\colon (\neg r_1(y,x) \text{ or } \neg\forall y\colon (\neg r_1(x,y) \text{ or } \forall x\colon (\neg r_1(y,x) \text{ or } p_3(x))$ o

LCL690∧1.p Prove K in the CS4 translation
include('Axioms/LCL008^0.ax')
include('Axioms/LCL009^0.ax')
∀a: \$i → \$o, b: \$i → \$o: (cs4_valid@(cs4_impl@(cs4_box@(cs4_impl@(cs4_atom@a)@(cs4_atom@b)))@(cs4_impl@(cs4_box@(cs4

LCL691∧1.p Prove reflexivity in the CS4 translation
include('Axioms/LCL008^0.ax')
include('Axioms/LCL009^0.ax')
∀a: \$i → \$o: (cs4_valid@(cs4_impl@(cs4_box@(cs4_atom@a))@(cs4_atom@a))) thf(cs4_refl, conjecture)

LCL692∧1.p Prove transitivity in the CS4 translation
include('Axioms/LCL008^0.ax')
include('Axioms/LCL009^0.ax')
∀a: \$i → \$o: (cs4_valid@(cs4_impl@(cs4_box@(cs4_atom@a))@(cs4_box@(cs4_box@(cs4_atom@a))))) thf(cs4_trans, conject

LCL693∧1.p Prove the Barcan formula in the CS4 translation
include('Axioms/LCL008^0.ax')
include('Axioms/LCL009^0.ax')
∀a: individuals → \$i → \$o: (cs4_valid@(cs4_impl@(cs4_all@λi: individuals: (cs4_box@(cs4_atom@(a@i))))@(cs4_box@(cs4_all@

LCL694∧1.p Prove the converse Barcan formula in the CS4 translation
include('Axioms/LCL008^0.ax')
include('Axioms/LCL009^0.ax')
∀a: individuals → \$i → \$o: (cs4_valid@(cs4_impl@(cs4_box@(cs4_all@λi: individuals: (cs4_atom@(a@i))))@(cs4_all@λi: indivi

LCL695∧1.p Propositional intuitionistic logic in HOL
An embedding of propositional intuitionisitc logic in HOL based on Goedel's first translation of propositional intuitionistic logic to modal logic S4.
include('Axioms/LCL010^0.ax')

LCL696∧1.p Propositional intuitionistic logic in HOL
An embedding of propositional intuitionisitc logic in HOL based on Goedel's second translation of propositional intuitionistic logic to modal logic S4.
include('Axioms/LCL011^0.ax')

LCL697∧1.p Propositional intuitionistic logic in HOL
An embedding of propositional intuitionisitc logic in HOL based on the McKinsey/Tarski translation of propositional intuitionistic logic to modal logic S4.
include('Axioms/LCL012^0.ax')

LCL698∧1.p Embedding of quantified multimodal logic in simple type theory
include('Axioms/LCL013^0.ax')

LCL699∧1.p Accessibility relation implies axiom for reflexivity
include('Axioms/LCL013^0.ax')
∀r: \$i → \$i → \$o: ((mreflexive@r) ⇒ (mvalid@(mforall_prop@λa: \$i → \$o: (mimplies@(mbox@r@a)@a)))) thf(conj, conj

LCL700∧1.p Accessibility relation implies axiom for symmetry
include('Axioms/LCL013^0.ax')
∀r: \$i → \$i → \$o: ((msymmetric@r) ⇒ (mvalid@(mforall_prop@λa: \$i → \$o: (mimplies@a@(mbox@r@(mdia@r@a))))))

LCL701∧1.p Accessibility relation implies axiom for seriality
include('Axioms/LCL013^0.ax')
∀r: \$i → \$i → \$o: ((mserial@r) ⇒ (mvalid@(mforall_prop@λa: \$i → \$o: (mimplies@(mbox@r@a)@(mdia@r@a))))) thf(c

LCL702∧1.p Accessibility relation implies axiom for transitivity
include('Axioms/LCL013^0.ax')
∀r: \$i → \$i → \$o: ((mtransitive@r) ⇒ (mvalid@(mforall_prop@λa: \$i → \$o: (mimplies@(mbox@r@a)@(mbox@r@(mbox@r@

LCL703∧1.p Accessibility relation implies axiom for Euclidianity
include('Axioms/LCL013^0.ax')
∀r: \$i → \$i → \$o: ((meuclidean@r) ⇒ (mvalid@(mforall_prop@λa: \$i → \$o: (mimplies@(mdia@r@a)@(mbox@r@(mdia@r@a

LCL704∧1.p Accessibility relation implies axiom for partial functionality
include('Axioms/LCL013^0.ax')

∀r: $i → $i → $o: ((mpartially_functional@r) ⇒ (mvalid@(mforall_prop@λa: $i → $o: (mimplies@(mdia@r@a)@(mbox@r@a

LCL705^1.p Accessibility relation implies axiom for functionality
include('Axioms/LCL013^0.ax')
∀r: $i → $i → $o: ((mfunctional@r) ⇒ (mvalid@(mforall_prop@λa: $i → $o: (mequiv@(mdia@r@a)@(mbox@r@a))))) th

LCL706^1.p Accessibility relation implies axiom for weak density
include('Axioms/LCL013^0.ax')
∀r: $i → $i → $o: ((mweakly_dense@r) ⇒ (mvalid@(mforall_prop@λa: $i → $o: (mimplies@(mbox@r@(mbox@r@a))@(mbox

LCL707^1.p Accessibility relation implies axiom for weak connectedness
include('Axioms/LCL013^0.ax')
∀r: $i → $i → $o: ((mweakly_connected@r) ⇒ (mvalid@(mforall_prop@λa: $i → $o: (mforall_prop@λb: $i → $o: (mor@(mbox@r@(mimplies@(mand@a@(mbox@r@a))@b))@(mbox@r@(mimplies@(mand@b@(mbox@r@b))@a)))))))

LCL708^1.p Accessibility relation implies axiom for weak directedness
include('Axioms/LCL013^0.ax')
∀r: $i → $i → $o: ((mweakly_directed@r) ⇒ (mvalid@(mforall_prop@λa: $i → $o: (mimplies@(mdia@r@(mbox@r@a))@(mb

LCL709^1.p Axiom implies accessibility relation for reflexivity
include('Axioms/LCL013^0.ax')
∀r: $i → $i → $o: ((mvalid@(mforall_prop@λa: $i → $o: (mimplies@(mbox@r@a)@a))) ⇒ (mreflexive@r)) thf(conj, conj

LCL710^1.p Axiom implies accessibility relation for symmetry
include('Axioms/LCL013^0.ax')
∀r: $i → $i → $o: ((mvalid@(mforall_prop@λa: $i → $o: (mimplies@a@(mbox@r@(mdia@r@a))))) ⇒ (msymmetric@r))

LCL711^1.p Axiom implies accessibility relation for seriality
include('Axioms/LCL013^0.ax')
∀r: $i → $i → $o: ((mvalid@(mforall_prop@λa: $i → $o: (mimplies@(mbox@r@a)@(mdia@r@a)))) ⇒ (mserial@r)) thf(c

LCL712^1.p Axiom implies accessibility relation for transitivity
include('Axioms/LCL013^0.ax')
∀r: $i → $i → $o: ((mvalid@(mforall_prop@λa: $i → $o: (mimplies@(mbox@r@a)@(mbox@r@(mbox@r@a))))) ⇒ (mtransitive@r)) thf(conj, conjecture)

LCL713^1.p Axiom implies accessibility relation for Euclidianity
include('Axioms/LCL013^0.ax')
∀r: $i → $i → $o: ((mvalid@(mforall_prop@λa: $i → $o: (mimplies@(mdia@r@a)@(mbox@r@(mdia@r@a))))) ⇒ (meuclidean@r)) thf(conj, conjecture)

LCL714^1.p Axiom implies accessibility relation for partial functionality
include('Axioms/LCL013^0.ax')
∀r: $i → $i → $o: ((mvalid@(mforall_prop@λa: $i → $o: (mimplies@(mdia@r@a)@(mbox@r@a)))) ⇒ (mpartially_functional

LCL715^1.p Axiom implies accessibility relation for functionality
include('Axioms/LCL013^0.ax')
∀r: $i → $i → $o: ((mvalid@(mforall_prop@λa: $i → $o: (mequiv@(mdia@r@a)@(mbox@r@a)))) ⇒ (mfunctional@r)) th

LCL716^1.p Axiom implies accessibility relation for weak density
include('Axioms/LCL013^0.ax')
∀r: $i → $i → $o: ((mvalid@(mforall_prop@λa: $i → $o: (mimplies@(mbox@r@(mbox@r@a))@(mbox@r@a)))) ⇒ (mweakly_dense@r)) thf(conj, conjecture)

LCL717^1.p Axiom implies accessibility relation for weak connectedness
include('Axioms/LCL013^0.ax')
∀r: $i → $i → $o: ((mvalid@(mforall_prop@λa: $i → $o: (mforall_prop@λb: $i → $o: (mor@(mbox@r@(mimplies@(mand@a@
(mweakly_connected@r)) thf(conj, conjecture)

LCL718^1.p Axiom implies accessibility relation for weak directedness
include('Axioms/LCL013^0.ax')
∀r: $i → $i → $o: ((mvalid@(mforall_prop@λa: $i → $o: (mimplies@(mdia@r@(mbox@r@a))@(mbox@r@(mdia@r@a))))) ⇒
(mweakly_directed@r)) thf(conj, conjecture)

LCL719^1.p Necessitation rule holds in monomodal logic K
include('Axioms/LCL013^0.ax')
include('Axioms/LCL013^1.ax')
phi: $i → $o thf(phi, type)
(mvalid@phi) ⇒ (mvalid@(mbox_k@phi)) thf(conj, conjecture)

LCL720^1.p Distribution axiom holds in monomodal logic K
include('Axioms/LCL013^0.ax')
include('Axioms/LCL013^1.ax')
phi: $i → $o thf(phi, type)
psi: $i → $o thf(psi, type)
mvalid@(mimplies@(mbox_k@(mimplies@phi@psi))@(mimplies@(mbox_k@phi)@(mbox_k@psi))) thf(conj, conjecture)

LCL721^1.p Axiom D holds in monomodal logic D
include('Axioms/LCL013^0.ax')
include('Axioms/LCL013^2.ax')
phi: $i → $o thf(phi, type)
mvalid@(mimplies@(mbox_d@phi)@(mdia_d@phi)) thf(conj, conjecture)

LCL722^1.p Axiom M holds in monomodal logic M
include('Axioms/LCL013^0.ax')
include('Axioms/LCL013^3.ax')
phi: $i → $o thf(phi, type)
mvalid@(mimplies@(mbox_m@phi)@phi) thf(conj, conjecture)

LCL723^1.p Axioms M and B hold in monomodal logic B
include('Axioms/LCL013^0.ax')
include('Axioms/LCL013^4.ax')
phi: $i → $o thf(phi, type)
mvalid@(mimplies@(mbox_b@phi)@phi) and mvalid@(mimplies@phi@(mbox_b@(mdia_b@phi))) thf(conj, conjecture)

LCL724^1.p Axioms M and 4 hold in monomodal logic S4
include('Axioms/LCL013^0.ax')
include('Axioms/LCL013^5.ax')
phi: $i → $o thf(phi, type)
mvalid@(mimplies@(mbox_s$_{4}$@phi)@phi) and mvalid@(mimplies@(mbox_s$_{4}$@phi)@(mbox_s$_{4}$@(mbox_s$_{4}$@phi))) thf(conj, c

LCL725^1.p Axioms M, 4, and B hold in monomodal logic S5
include('Axioms/LCL013^0.ax')
include('Axioms/LCL013^6.ax')
phi: $i → $o thf(phi, type)
mvalid@(mimplies@(mbox_s$_{5}$@phi)@phi) and mvalid@(mimplies@(mbox_s$_{5}$@phi)@(mbox_s$_{5}$@(mbox_s$_{5}$@phi))) and mvalid@

LCL726^5.p TPS problem THM534
AC1 => AC17 from [RR93].
a: $tType thf(a_type, type)
∀xs: (*a* → $o) → $o: (∀*x*: *a* → $o: ((xs@*x*) ⇒ ∃xt: *a*: (*x*@xt)) ⇒ ∃xf: (*a* → $o) → *a*: ∀*x*: *a* → $o: ((xs@*x*) ⇒ (*x*@(xf@*x*)))) ⇒ ∀xg: ((*a* → $o) → *a*) → *a* → $o: (∀xh: (*a* → $o) → *a*: ∃xu: *a*: (xg@xh@xu) ⇒ ∃xf: (*a* → $o) → *a*: (xg@xf@(xf@(xg@xf)))) thf(cTHM$_{534}$, conjecture)

LCL727^5.p TPS problem THM533
AC3 => AC1 from [RR93].
a: $tType thf(a_type, type)
∀xr: (*a* → $o) → *a* → $o: ∃xg: (*a* → $o) → *a*: ∀xx: *a* → $o: (∃xy: *a*: (xr@xx@xy) ⇒ (xr@xx@(xg@xx))) ⇒ ∀xs: (*a* → $o) → $o: (∀*x*: *a* → $o: ((xs@*x*) ⇒ ∃xt: *a*: (*x*@xt)) ⇒ ∃xf: (*a* → $o) → *a*: ∀*x*: *a* → $o: ((xs@*x*) ⇒ (*x*@(xf@*x*)))) thf(cTHM$_{533}$, conjecture)

LCL728^5.p TPS problem THM532
AC1 => AC3 from [RR93].
b: $tType thf(b_type, type)
a: $tType thf(a_type, type)
∀xs: (*b* → $o) → $o: (∀*x*: *b* → $o: ((xs@*x*) ⇒ ∃xy: *b*: (*x*@xy)) ⇒ ∃xf: (*b* → $o) → *b*: ∀*x*: *b* → $o: ((xs@*x*) ⇒ (*x*@(xf@*x*)))) ⇒ ∀xr: *a* → *b* → $o: ∃xg: *a* → *b*: ∀xx: *a*: (∃xy: *b*: (xr@xx@xy) ⇒ (xr@xx@(xg@xx))) thf(cTHM$_{532}$, conject

LCL729^5.p TPS problem THM560
AC1(A) equiv AC3(OA,A) from [RR93]. Note that AC3 usually refers to 'relations' where here we are using it for relations on OA x A.
a: $tType thf(a_type, type)
∀xr: (*a* → $o) → *a* → $o: ∃xg: (*a* → $o) → *a*: ∀xx: *a* → $o: (∃xy: *a*: (xr@xx@xy) ⇒ (xr@xx@(xg@xx))) ⟺ ∀xs: (*a* → $o) → $o: (∀*x*: *a* → $o: ((xs@*x*) ⇒ ∃xt: *a*: (*x*@xt)) ⇒ ∃xf: (*a* → $o) → *a*: ∀*x*: *a* → $o: ((xs@*x*) ⇒ (*x*@(xf@*x*)))) thf(cTHM$_{560}$, conjecture)

LCL730^5.p TPS problem X5310
Related to the axiom of choice.
b: $tType   thf(b_type, type)
∀xr: (*b* → $o) → *b* → $o: (∀xx: *b* → $o: ∃xy: *b*: (xr@xx@xy) ⇒ ∃xf: (*b* → $o) → *b*: ∀xx: *b* → $o: (xr@xx@(xf@xx))) ⇒ ∃xj: (*b* → $o) → *b*: ∀xp: *b* → $o: (∃xz: *b*: (xp@xz) ⇒ (xp@(xj@xp)))   thf(cX_{5310}, conjecture)

LCL731^5.p TPS problem THM541
Equivalence of global choice at type A (usual way of expressing AC in type theory).
a: $tType   thf(a_type, type)
∃xf: (*a* → $o) → *a*: ∀*x*: *a* → $o: (∃xt: *a*: (*x*@xt) ⇒ (*x*@(xf@*x*))) ⟺ ∀xs: (*a* → $o) → $o: (∀*x*: *a* → $o: ((xs@*x*) ⇒ ∃xt: *a*: (*x*@xt)) ⇒ ∃xf: (*a* → $o) → *a*: ∀*x*: *a* → $o: ((xs@*x*) ⇒ (*x*@(xf@*x*))))   thf(cTHM_{541}, conjecture)

LCL732^5.p TPS problem from AC-THMS
Related to the axiom of choice.
b: $tType   thf(b_type, type)
y: $i   thf(*y*, type)
p: $i → $o   thf(*p*, type)
∀xx: *b* → $o: $\exists \mathrm{xy}_0$: *b*: ($\exists \mathrm{xx}_0$: $i: (*p*@$\mathrm{xx}_0$) ⇒ (*p*@*y*)) ⇒ ∃xf: (*b* → $o) → *b*: ∀xx: *b* → $o: ($\exists \mathrm{xx}_0$: $i: (*p*@$\mathrm{xx}_0$) ⇒ (*p*@*y*))   thf(cX5310_SUB_3, conjecture)

LCL733^5.p TPS problem from AC-THMS
Related to the axiom of choice.
b: $tType   thf(b_type, type)
∀xr: (*b* → $o) → *b* → $o: (∀xx: *b* → $o: ∃xy: *b*: ($\exists \mathrm{xx}_0$: *b*: (xx@$\mathrm{xx}_0$) ⇒ (xx@xy)) ⇒ ∃xf: (*b* → $o) → *b*: ∀xx: *b* → $o: ($\exists \mathrm{xx}_0$: *b*: (xx@$\mathrm{xx}_0$) ⇒ (xx@(xf@xx)))) ⇒ ∃xj: (*b* → $o) → *b*: ∀xp: *b* → $o: (∃xz: *b*: (xp@xz) ⇒ (xp@(xj@xp)))   thf(cX5310_SUB_4, conjecture)

LCL734^5.p TPS problem from AC-THMS
Related to the axiom of choice.
b: $tType   thf(b_type, type)
$\forall \mathrm{xr}_3$: (*b* → $o) → *b* → $o, xr_4: (*b* → $o) → *b* → $o: (∀xx: *b* → $o: ∃xy: *b*: ($\mathrm{xr}_3$@xx@xy or xr_4@xx@xy) ⇒ ∃xf: (*b* → $o) → *b*: ∀xx: *b* → $o: ($\mathrm{xr}_3$@xx@(xf@xx) or xr_4@xx@(xf@xx))) ⇒ ∃xj: (*b* → $o) → *b*: ∀xp: *b* → $o: (∃xz: *b*: (xp@xz) ⇒ (xp@(xj@xp)))   thf(cX5310B, conjecture)

LCL735^5.p TPS problem from AC-THMS
Related to the axiom of choice.
b: $tType   thf(b_type, type)
∀xa: (*b* → $o) → *b* → *b* → $o, xb: (*b* → $o) → *b* → *b* → $o: (∀xx: *b* → $o: ∃xy: *b*: ∀xw: *b*: (xa@xx@xy@xw or xb@xx@xy@xw)
∃xf: (*b* → $o) → *b*: ∀xx: *b* → $o, xw: *b*: (xa@xx@(xf@xx)@xw or xb@xx@(xf@xx)@xw)) ⇒ ∃xj: (*b* → $o) → *b*: ∀xp: *b* → $o: (∃xz: *b*: (xp@xz) ⇒ (xp@(xj@xp)))   thf(cX5310_SUB_5, conjecture)

LCL736^5.p TPS problem from AC-THMS
Related to the axiom of choice.
b: $tType   thf(b_type, type)
$\forall \mathrm{xr}_{41}$: (*b* → $o) → *b* → *b* → $o, xr_{42}: (*b* → $o) → *b* → *b* → $o: (∀xx: *b* → $o: ∃xy: *b*, xw_{11}: *b*: (xr_{41}@xx@xy@xw_{11} or xr_{42}@xx@
∃xf: (*b* → $o) → *b*: ∀xx: *b* → $o: $\exists \mathrm{xw}_{11}$: *b*: (xr_{41}@xx@(xf@xx)@xw_{11} or xr_{42}@xx@(xf@xx)@xw_{11})) ⇒ ∃xj: (*b* → $o) → *b*: ∀xp: *b* → $o: (∃xz: *b*: (xp@xz) ⇒ (xp@(xj@xp)))   thf(cX5310D, conjecture)

LCL738^5.p TPS problem from AC-THMS
Related to the axiom of choice.
b: $tType   thf(b_type, type)
$\exists \mathrm{xr}_{24}$: (*b* → $o) → *b* → *b* → $o, xr_{23}: (*b* → $o) → *b* → *b* → $o: (∀xx: *b* → $o: ∃xy: *b*: $\forall \mathrm{xw}_6$: *b*: (xr_{23}@xx@xy@xw_6 or xr_{24}@xx@x
∃xf: (*b* → $o) → *b*: ∀xx: *b* → $o, xw_6: *b*: (xr_{23}@xx@(xf@xx)@xw_6 or xr_{24}@xx@(xf@xx)@xw_6)) ⇒ ∃xj: (*b* → $o) → *b*: ∀xp: *b* → $o: (∃xz: *b*: (xp@xz) ⇒ (xp@(xj@xp)))   thf(cX5310_SUB, conjecture)

LCL739^5.p TPS problem from AC-THMS
Related to the axiom of choice.
b: $tType   thf(b_type, type)
$\forall \mathrm{xr}_{24}$: (*b* → $o) → *b* → *b* → $o, xr_{23}: (*b* → $o) → *b* → *b* → $o: (∀xx: *b* → $o: ∃xy: *b*: $\forall \mathrm{xw}_6$: *b*: (xr_{23}@xx@xy@xw_6 or xr_{24}@xx@x
∃xf: (*b* → $o) → *b*: ∀xx: *b* → $o, xw_6: *b*: (xr_{23}@xx@(xf@xx)@xw_6 or xr_{24}@xx@(xf@xx)@xw_6)) ⇒ ∃xj: (*b* → $o) → *b*: ∀xp: *b* → $o: (∃xz: *b*: (xp@xz) ⇒ (xp@(xj@xp)))   thf(cX5310_SUB_2, conjecture)

LCL740^5.p TPS problem from AC-THMS
Related to the axiom of choice.
b: $tType   thf(b_type, type)

∀xr_{12}: (*b* → \$o) → *b* → *b* → \$o, xr_{13}: (*b* → \$o) → *b* → *b* → \$o: (∃xf: (*b* → \$o) → *b*: ∀xx: *b* → \$o, xw_4: *b*: (xr_{12}@xx@(xf@xx)@xw
\$o: ∃xy: *b*: ∀xw_4: *b*: (xr_{12}@xx@xy@xw_4 or xr_{13}@xx@xy@xw_4)) ⇒ ∃xj: (*b* → \$o) → *b*: ∀xp: *b* → \$o: (∃xz: *b*: (xp@xz) ⇒ (xp@(xj@xp))) thf(cX5310_SUB_R, conjecture)

LCL741^5.p TPS problem from AC-THMS
Related to the axiom of choice.
b: \$tType thf(b_type, type)
∀xr: (*b* → \$o) → *b* → \$o, xr_{27}: (*b* → \$o) → *b* → *b* → \$o, xr_{28}: (*b* → \$o) → *b* → *b* → \$o: (∀xx: *b* →
\$o: ∃xy: *b*: ∀xw_{11}: *b*: (xr_{27}@xx@xy@xw_{11} or xr_{28}@xx@xy@xw_{11}) ⇒ ∃xf: (*b* → \$o) → *b*: ∀xx: *b* → \$o, xw_{11}: *b*: (xr_{27}@xx@(xf@x
∃xj: (*b* → \$o) → *b*: ∀xp: *b* → \$o: (∃xz: *b*: (xp@xz) ⇒ (xp@(xj@xp))) thf(cX5310X_pme, conjecture)

LCL742^5.p TPS problem from AC-FUNS-THMS
f: \$i → \$i thf(*f*, type)
g: \$i → \$i thf(*g*, type)
∃xj: (\$i → \$o) → \$i: ∀xp: \$i → \$o: (∃xx: \$i: (xp@xx) ⇒ (xp@(xj@xp))) ⇒ (∀xx: \$i, xy: \$i: ((*g*@xx) = (*g*@xy) ⇒
(*f*@xx) = (*f*@xy)) ⇒ ∃xh: \$i → \$i: (λxx: \$i: (xh@(*g*@xx))) = *f*) thf(cTHM588AC_pme, conjecture)

LCL743^5.p TPS problem from AXIOMOFDESCR
Related to the axiom of description.
a: \$tType thf(a_type, type)
cF: *a* → \$o thf(cF, type)
cJ: (*a* → \$o) → *a* thf(cJ, type)
cX: *a* thf(cX, type)
(cF@cX) ⇒ (∀*y*: *a*: ((cF@*y*) ⇒ cX = *y*) ⇒ (cF@(cJ@cF))) thf(cDESCR_CHURCH, conjecture)

LCL744-1.p Strong normalization of typed lambda calculus 019_3
c_Lambda_Obeta(v_s, v_t) ⇒ c_Lambda_Obeta(c_Lambda_OdB_OAbs(v_s), c_Lambda_OdB_OAbs(v_t)) cnf(cls_abs_0, axic
c_Lambda_Obeta(c_Lambda_OdB_OAbs(v_r), v_s) ⇒ v_s = c_Lambda_OdB_OAbs(c_Lambda_Osko__Lambda__Xbeta__cases__
c_Lambda_Obeta(v_s, v_t) ⇒ c_Lambda_Obeta(c_Lambda_OdB_OApp(v_s, v_u), c_Lambda_OdB_OApp(v_t, v_u)) cnf(cls_
c_Lambda_Obeta(v_s, v_t) ⇒ c_Lambda_Obeta(c_Lambda_OdB_OApp(v_u, v_s), c_Lambda_OdB_OApp(v_u, v_t)) cnf(cls_
c_Lambda_Obeta(v_r, v_s) ⇒ c_Lambda_Obeta(c_Lambda_Osubst(v_r, v_t, v_i), c_Lambda_Osubst(v_s, v_t, v_i)) cnf(cls_su
c_Lambda_OdB_OAbs(v_dB) = c_Lambda_OdB_OAbs(v_dB_H) ⇒ v_dB = v_dB_H cnf($cls\_dB\_Osimps\_I3\_J_0$, axiom)
c_Lambda_Obeta(c_Lambda_OdB_OApp(c_Lambda_OdB_OAbs(v_s), v_t), c_Lambda_Osubst(v_s, v_t, c_HOL_Ozero__class_Oze
c_Lambda_OdB_OApp(v_dB_1, v_dB_2) = c_Lambda_OdB_OApp(v_dB1_H, v_dB2_H) ⇒ v_dB_2 = v_dB2_H cnf(cls_dB_Osir
c_Lambda_OdB_OApp(v_dB_1, v_dB_2) = c_Lambda_OdB_OApp(v_dB1_H, v_dB2_H) ⇒ v_dB_1 = v_dB1_H cnf(cls_dB_Osir
c_Lambda_Obeta(c_Lambda_OdB_OAbs(v_r), v_s) ⇒ c_Lambda_Obeta(v_r, $c\_Lambda\_Osko\_\_Lambda\_\_Xbeta\_\_cases\_\_2\_{1}$(v_r
c_Lambda_OdB_OAbs(v_dB_H) ≠ c_Lambda_OdB_OApp(v_dB_1, v_dB_2) cnf($cls\_dB\_Osimps\_I9\_J_0$, axiom)
c_Lambda_OdB_OApp(v_dB_1, v_dB_2) ≠ c_Lambda_OdB_OAbs(v_dB_H) cnf($cls\_dB\_Osimps\_I8\_J_0$, axiom)
c_Lambda_Osubst(c_Lambda_OdB_OApp(v_t, v_u), v_s, v_k) = c_Lambda_OdB_OApp(c_Lambda_Osubst(v_t, v_s, v_k), c_Lamb
c_Lambda_Olift(c_Lambda_OdB_OApp(v_s, v_t), v_k) = c_Lambda_OdB_OApp(c_Lambda_Olift(v_s, v_k), c_Lambda_Olift(v_t,
c_InductTermi_OIT(v_r) ⇒ c_InductTermi_OIT(c_Lambda_OdB_OAbs(v_r)) cnf(cls_Lambda_0, axiom)
c_Lambda_Osubst(c_Lambda_Olift(v_t, v_k), v_s, v_k) = v_t cnf($cls\_subst\_\_lift_0$, axiom)
c_Lambda_Obeta(v_r, v_s) ⇒ c_Lambda_Obeta(c_Lambda_Olift(v_r, v_i), c_Lambda_Olift(v_s, v_i)) cnf(cls_lift__preserves_
c_InductTermi_OIT(v_t) cnf($cls\_conjecture_0$, negated_conjecture)
¬ c_InductTermi_OIT(c_Lambda_Olift(v_t, v_i)) cnf($cls\_conjecture_1$, negated_conjecture)

LCL859^1.p Modal logic S5(=M5) coincides with MB5
include('Axioms/LCL013^0.ax')
∀*r*: \$i → \$i → \$o: ((mreflexive@*r* and meuclidean@*r*) ⟺ (mreflexive@*r* and msymmetric@*r* and meuclidean@*r*)) thf(c

LCL860^1.p Modal logic S5(=M5) coincides with M4B5
include('Axioms/LCL013^0.ax')
∀*r*: \$i → \$i → \$o: ((mreflexive@*r* and meuclidean@*r*) ⟺ (mreflexive@*r* and mtransitive@*r* and msymmetric@*r* and meucl

LCL861^1.p Modal logic S5(=M5) coincides with M45
include('Axioms/LCL013^0.ax')
∀*r*: \$i → \$i → \$o: ((mreflexive@*r* and meuclidean@*r*) ⟺ (mreflexive@*r* and mtransitive@*r* and meuclidean@*r*)) thf(co

LCL862^1.p Modal logic S5(=M5) coincides with M4B
include('Axioms/LCL013^0.ax')
∀*r*: \$i → \$i → \$o: ((mreflexive@*r* and meuclidean@*r*) ⟺ (mreflexive@*r* and mtransitive@*r* and msymmetric@*r*)) thf(c

LCL863^1.p Modal logic S5(=M5) coincides with D4B
include('Axioms/LCL013^0.ax')
∀*r*: \$i → \$i → \$o: ((mreflexive@*r* and meuclidean@*r*) ⟺ (mserial@*r* and mtransitive@*r* and msymmetric@*r*)) thf(conj

LCL864^1.p Modal logic S5(=M5) coincides with D4B5
include('Axioms/LCL013^0.ax')
∀*r*: \$i → \$i → \$o: ((mreflexive@*r* and meuclidean@*r*) ⟺ (mserial@*r* and mtransitive@*r* and msymmetric@*r* and meuclide

LCL865^1.p Modal logic S5(=M5) coincides with DB5
include('Axioms/LCL013^0.ax')
∀*r*: \$i → \$i → \$o: ((mreflexive@*r* and meuclidean@*r*) ⟺ (mserial@*r* and msymmetric@*r* and meuclidean@*r*)) thf(conj

LCL866^1.p Modal logic KB5 coincides with K4B5
include('Axioms/LCL013^0.ax')
∀*r*: \$i → \$i → \$o: ((msymmetric@*r* and meuclidean@*r*) ⟺ (mtransitive@*r* and msymmetric@*r* and meuclidean@*r*)) t

LCL867^1.p Modal logic KB5 coincides with K4B
include('Axioms/LCL013^0.ax')
∀*r*: \$i → \$i → \$o: ((msymmetric@*r* and meuclidean@*r*) ⟺ (mtransitive@*r* and msymmetric@*r*)) thf(conj, conjecture)

LCL868^1.p Modal logic D45 'includes' modal logic M5
include('Axioms/LCL013^0.ax')
∀*r*: \$i → \$i → \$o: ((mreflexive@*r* and meuclidean@*r*) ⇒ (mserial@*r* and mtransitive@*r* and meuclidean@*r*)) thf(conj, co

LCL869^1.p Modal logic M5 'includes' modal logic D45
include('Axioms/LCL013^0.ax')
∀*r*: \$i → \$i → \$o: ((mserial@*r* and mtransitive@*r* and meuclidean@*r*) ⇒ (mreflexive@*r* and meuclidean@*r*)) thf(conj, co

LCL870^1.p The Barcan formula is valid in quantified modal logic K
include('Axioms/LCL013^0.ax')
r: \$i → \$i → \$o thf(*r*, type)
p: mu → \$i → \$o thf(*p*, type)
mvalid@(mimplies@(mforall_ind@λ*x*: mu: (mbox@*r*@(*p*@*x*)))@(mbox@*r*@(mforall_ind@λ*x*: mu: (*p*@*x*)))) thf(ex_1, conject

LCL871^1.p The converse Barcan formula is valid in quantified modal logic K
include('Axioms/LCL013^0.ax')
r: \$i → \$i → \$o thf(*r*, type)
p: mu → \$i → \$o thf(*p*, type)
mvalid@(mimplies@(mbox@*r*@(mforall_ind@λ*x*: mu: (*p*@*x*)))@(mforall_ind@λ*x*: mu: (mbox@*r*@(*p*@*x*)))) thf(ex2b, conjec

LCL872^1.p Correspondence between box and diamond and a confluence property
include('Axioms/LCL013^0.ax')
i: \$i → \$i → \$o thf(*i*, type)
j: \$i → \$i → \$o thf(*j*, type)
k: \$i → \$i → \$o thf(*k*, type)
l: \$i → \$i → \$o thf(*l*, type)
confluence: (\$i → \$i → \$o) → (\$i → \$i → \$o) → (\$i → \$i → \$o) → (\$i → \$i → \$o) → \$o thf(confluences_type, type)
confluence = (λ*i*: \$i → \$i → \$o, *j*: \$i → \$i → \$o, *k*: \$i → \$i → \$o, *l*: \$i → \$i → \$o: ∀*a*: \$i, *b*: \$i, *c*: \$i: ((*i*@*a*@*b* and *k*@*a*@*c*) ⇒
∃*d*: \$i: (*j*@*b*@*d* and *l*@*c*@*d*))) thf(confluence, definition)
(mvalid@(mforall_prop@λ*p*: \$i → \$o: (mimplies@(mdia@*i*@(mbox@*j*@*p*))@(mbox@*k*@(mdia@*l*@*p*))))) ⟺ (confluence@*i*@

LCL873^1.p Commutativity implies orthogonality in 2-D modal logic S5
include('Axioms/LCL013^0.ax')
ra: \$i → \$i → \$o thf(ra, type)
rb: \$i → \$i → \$o thf(rb, type)
mreflexive@ra thf(ax_1, axiom)
mreflexive@rb thf(ax_2, axiom)
mtransitive@ra thf(ax_3, axiom)
mtransitive@rb thf(ax_4, axiom)
meuclidean@ra thf(ax_5, axiom)
meuclidean@rb thf(ax_6, axiom)
mvalid@(mforall_prop@λ*a*: \$i → \$o: (mforall_prop@λ*b*: \$i → \$o: (mimplies@(mbox@ra@(mor@(mbox@ra@*a*)@(mbox@rb@*b*))
\$o: (mforall_prop@λ*b*: \$i → \$o: (mimplies@(mbox@rb@(mor@(mbox@ra@*a*)@(mbox@rb@*b*)))@(mor@(mbox@rb@*a*)@(mbox

LCL874^1.p Inclusion statement in a 2-D logic of knowledge and belief
Suppose we want to work with a 2-dimensional logic combining a modality box_k of knowledge with a modality box_b of belief. Moreover, suppose we model box_k as an S5 modality and box_b as an D45 modality and let us furthermore add two axioms characterizing their relationship. We may then want to check whether or not box_k and box_b coincide, i.e., whether box_k includes box_b

include('Axioms/LCL013^0.ax')
rk: \$i → \$i → \$o   thf(rk, type)
rb: \$i → \$i → \$o   thf(rb, type)
mreflexive@rk   thf(ax_1, axiom)
mserial@rb   thf(ax_2, axiom)
mtransitive@rk   thf(ax_3, axiom)
mtransitive@rb   thf(ax_4, axiom)
meuclidean@rk   thf(ax_5, axiom)
meuclidean@rb   thf(ax_6, axiom)
mvalid@(mforall_prop@λa: \$i → \$o: (mimplies@(mbox@rk@a)@(mbox@rb@a)))   thf(ax_7, axiom)
mvalid@(mforall_prop@λa: \$i → \$o: (mimplies@(mbox@rb@a)@(mbox@rb@(mbox@rk@a))))   thf(ax_8, axiom)
mvalid@(mforall_prop@λa: \$i → \$o: (mimplies@(mbox@rb@a)@(mbox@rk@a)))   thf(conj, conjecture)

LCL875-1.p The Rezus formula
(is_a_theorem(implies(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y)   cnf(condensed_detachment, axiom)
is_a_theorem(implies(implies(a, b), implies(implies(c, a), implies(c, b))))   cnf(b, axiom)
is_a_theorem(implies(implies(a, implies(b, c)), implies(b, implies(a, c))))   cnf(c, axiom)
is_a_theorem(implies(a, a))   cnf(i, axiom)
is_a_theorem(implies(implies(u, implies(u, v)), implies(u, v)))   cnf(w, axiom)
¬ is_a_theorem(implies(implies(implies(x, implies(implies(implies(y, y), implies(implies(z, z), implies(implies(u, u), implies(imp

LCL876+1.p Prove Mv5 from MV1–MV4
∀y, x: ((is_a_theorem(implies(x, y)) and is_a_theorem(x)) ⇒ is_a_theorem(y))   fof(cd, axiom)
∀y, x: is_a_theorem(implies(x, implies(y, x)))   fof(mv_1, axiom)
∀z, y, x: is_a_theorem(implies(implies(x, y), implies(implies(y, z), implies(x, z))))   fof(mv_2, axiom)
∀y, x: is_a_theorem(implies(implies(implies(x, y), y), implies(implies(y, x), x)))   fof(mv_3, axiom)
∀y, x: is_a_theorem(implies(implies(not(x), not(y)), implies(y, x)))   fof(mv_4, axiom)
∀y, x: is_a_theorem(implies(implies(implies(x, y), implies(y, x)), implies(y, x)))   fof(mv_5, conjecture)

LCL877^1.p Variants of axiom 5
include('Axioms/LCL013^0.ax')
∀r: \$i → \$i → \$o: ((mvalid@(mforall_prop@λphi: \$i → \$o: (mimplies@(mnot@(mbox@r@phi))@(mbox@r@(mnot@(mbox@r@
(mvalid@(mforall_prop@λphi: \$i → \$o: (mimplies@(mdia@r@phi)@(mbox@r@(mdia@r@phi))))))   thf(conj, conjecture)

LCL877^2.p Variants of axiom 5
include('Axioms/LCL013^0.ax')
∀r: \$i → \$i → \$o: ((mvalid@(mforall_prop@λphi: \$i → \$o: (mimplies@(mnot@(mbox@r@phi))@(mbox@r@(mnot@(mbox@r@
(meuclidean@r))   thf(conj, conjecture)

LCL878^1.p Correspondence for axiom I
include('Axioms/LCL013^0.ax')
∀i: \$i → \$i → \$o, j: \$i → \$i → \$o: ((mvalid@(mforall_prop@λphi: \$i → \$o: (mimplies@(mbox@i@phi)@(mbox@j@phi)))) ⟸
∀u: \$i, v: \$i: ((j@u@v) ⇒ (i@u@v)))   thf(conj, conjecture)

LCL879^1.p Correspondence for axiom 4s
include('Axioms/LCL013^0.ax')
∀i: \$i → \$i → \$o, j: \$i → \$i → \$o: ((mvalid@(mforall_prop@λphi: \$i → \$o: (mimplies@(mbox@i@phi)@(mbox@j@(mbox@i@p
∀u: \$i, v: \$i, w: \$i: ((j@u@v and i@v@w) ⇒ (i@u@w)))   thf(conj, conjecture)

LCL880^1.p Correspondence for axiom 5s
include('Axioms/LCL013^0.ax')
∀i: \$i → \$i → \$o, j: \$i → \$i → \$o: ((mvalid@(mforall_prop@λphi: \$i → \$o: (mimplies@(mnot@(mbox@i@phi))@(mbox@j@(m
∀u: \$i, v: \$i, w: \$i: ((j@u@v and i@u@w) ⇒ (i@v@w)))   thf(conj, conjecture)

LCL882+1.p An involutive pocrim that does not have unique halving
∀a, b, c: $a' +' b' +' c = a' +' b' +' c$ fof(sos_{01}, axiom)
∀a, b: $a' +' b = b' +' a$ fof(sos_{02}, axiom)
∀a: $a' +'$ '0' $= a$ fof(sos_{03}, axiom)
∀a: $a' >=' a$ fof(sos_{04}, axiom)
∀x_0, x_1, x_2: (($x_0' >=' x_1$ and $x_1' >=' x_2$) ⇒ $x_0' >=' x_2$)   fof(sos_{05}, axiom)
∀x_3, x_4: (($x_3' >=' x_4$ and $x_4' >=' x_3$) ⇒ $x_3 = x_4$)   fof(sos_{06}, axiom)
∀x_5, x_6, x_7: ($x_5' +' x_6' >=' x_7$ ⟺ $x_6' >=' x_5' ==>' x_7$)   fof(sos_{07}, axiom)
∀a: $a' >='$ '0'   fof(sos_{08}, axiom)
∀x_8, x_9, x_{10}: ($x_8' >=' x_9$ ⇒ $x_8' +' x_{10}' >=' x_9' +' x_{10}$)   fof($sos_{09}$, axiom)

$\forall x_{11}, x_{12}, x_{13}$: $(x'_{11} >=' x_{12} \Rightarrow x'_{12} ==>' x'_{13} >=' x'_{11} ==>' x_{13})$ fof(sos_{10}, axiom)
$\forall x_{14}, x_{15}, x_{16}$: $(x'_{14} >=' x_{15} \Rightarrow x'_{16} ==>' x'_{14} >=' x'_{16} ==>' x_{15})$ fof(sos_{11}, axiom)
$\forall a$: $a' +' \text{'1'} = \text{'1'}$ fof(sos_{12}, axiom)
$\forall a$: $a' ==>' \text{'1'}' ==>' \text{'1'} = a$ fof(sos_{13}, axiom)
$\forall x_{17}, x_{18}, x_{19}$: $((x_{17} = x'_{17} ==>' x_{18} \text{ and } x_{19} = x'_{19} ==>' x_{18}) \Rightarrow x_{17} = x_{19})$ fof(goals_{14}, conjecture)

LCL883+1.p An involutive pocrim that is not a hoop
$\forall a, b, c$: $a' +' b' +' c = a' +' b' +' c$ fof(sos_{01}, axiom)
$\forall a, b$: $a' +' b = b' +' a$ fof(sos_{02}, axiom)
$\forall a$: $a' +' \text{'0'} = a$ fof(sos_{03}, axiom)
$\forall a$: $a' >=' a$ fof(sos_{04}, axiom)
$\forall x_0, x_1, x_2$: $((x'_0 >=' x_1 \text{ and } x'_1 >=' x_2) \Rightarrow x'_0 >=' x_2)$ fof(sos_{05}, axiom)
$\forall x_3, x_4$: $((x'_3 >=' x_4 \text{ and } x'_4 >=' x_3) \Rightarrow x_3 = x_4)$ fof(sos_{06}, axiom)
$\forall x_5, x_6, x_7$: $(x'_5 +' x'_6 >=' x_7 \iff x'_6 >=' x'_5 ==>' x_7)$ fof(sos_{07}, axiom)
$\forall a$: $a' >=' \text{'0'}$ fof(sos_{08}, axiom)
$\forall x_8, x_9, x_{10}$: $(x'_8 >=' x_9 \Rightarrow x'_8 +' x'_{10} >=' x'_9 +' x_{10})$ fof(sos_{09}, axiom)
$\forall x_{11}, x_{12}, x_{13}$: $(x'_{11} >=' x_{12} \Rightarrow x'_{12} ==>' x'_{13} >=' x'_{11} ==>' x_{13})$ fof(sos_{10}, axiom)
$\forall x_{14}, x_{15}, x_{16}$: $(x'_{14} >=' x_{15} \Rightarrow x'_{16} ==>' x'_{14} >=' x'_{16} ==>' x_{15})$ fof(sos_{11}, axiom)
$\forall a$: $a' +' \text{'1'} = \text{'1'}$ fof(sos_{12}, axiom)
$\forall a$: $a' ==>' \text{'1'}' ==>' \text{'1'} = a$ fof(sos_{13}, axiom)
$\forall x_{17}, x_{18}$: $x'_{17} +' x'_{17} ==>' x_{18} = x'_{18} +' x'_{18} ==>' x_{17}$ fof(goals_{14}, conjecture)

LCL884+1.p A bounded hoop that is not an involutive pocrim
$\forall a, b, c$: $a' +' b' +' c = a' +' b' +' c$ fof(sos_{01}, axiom)
$\forall a, b$: $a' +' b = b' +' a$ fof(sos_{02}, axiom)
$\forall a$: $a' +' \text{'0'} = a$ fof(sos_{03}, axiom)
$\forall a$: $a' >=' a$ fof(sos_{04}, axiom)
$\forall x_0, x_1, x_2$: $((x'_0 >=' x_1 \text{ and } x'_1 >=' x_2) \Rightarrow x'_0 >=' x_2)$ fof(sos_{05}, axiom)
$\forall x_3, x_4$: $((x'_3 >=' x_4 \text{ and } x'_4 >=' x_3) \Rightarrow x_3 = x_4)$ fof(sos_{06}, axiom)
$\forall x_5, x_6, x_7$: $(x'_5 +' x'_6 >=' x_7 \iff x'_6 >=' x'_5 ==>' x_7)$ fof(sos_{07}, axiom)
$\forall a$: $a' >=' \text{'0'}$ fof(sos_{08}, axiom)
$\forall x_8, x_9, x_{10}$: $(x'_8 >=' x_9 \Rightarrow x'_8 +' x'_{10} >=' x'_9 +' x_{10})$ fof(sos_{09}, axiom)
$\forall x_{11}, x_{12}, x_{13}$: $(x'_{11} >=' x_{12} \Rightarrow x'_{12} ==>' x'_{13} >=' x'_{11} ==>' x_{13})$ fof(sos_{10}, axiom)
$\forall x_{14}, x_{15}, x_{16}$: $(x'_{14} >=' x_{15} \Rightarrow x'_{16} ==>' x'_{14} >=' x'_{16} ==>' x_{15})$ fof(sos_{11}, axiom)
$\forall a, b$: $a' +' a' ==>' b = b' +' b' ==>' a$ fof(sos_{12}, axiom)
$\forall a$: $a' +' \text{'1'} = \text{'1'}$ fof(sos_{13}, axiom)
$\forall x_{17}$: $x'_{17} ==>' \text{'1'}' ==>' \text{'1'} = x_{17}$ fof(goals_{14}, conjecture)

LCL885+1.p An involutive pocrim that is not a coop
$\forall a, b, c$: $a' +' b' +' c = a' +' b' +' c$ fof(sos_{01}, axiom)
$\forall a, b$: $a' +' b = b' +' a$ fof(sos_{02}, axiom)
$\forall a$: $a' +' \text{'0'} = a$ fof(sos_{03}, axiom)
$\forall a$: $a' >=' a$ fof(sos_{04}, axiom)
$\forall x_0, x_1, x_2$: $((x'_0 >=' x_1 \text{ and } x'_1 >=' x_2) \Rightarrow x'_0 >=' x_2)$ fof(sos_{05}, axiom)
$\forall x_3, x_4$: $((x'_3 >=' x_4 \text{ and } x'_4 >=' x_3) \Rightarrow x_3 = x_4)$ fof(sos_{06}, axiom)
$\forall x_5, x_6, x_7$: $(x'_5 +' x'_6 >=' x_7 \iff x'_6 >=' x'_5 ==>' x_7)$ fof(sos_{07}, axiom)
$\forall a$: $a' >=' \text{'0'}$ fof(sos_{08}, axiom)
$\forall x_8, x_9, x_{10}$: $(x'_8 >=' x_9 \Rightarrow x'_8 +' x'_{10} >=' x'_9 +' x_{10})$ fof(sos_{09}, axiom)
$\forall x_{11}, x_{12}, x_{13}$: $(x'_{11} >=' x_{12} \Rightarrow x'_{12} ==>' x'_{13} >=' x'_{11} ==>' x_{13})$ fof(sos_{10}, axiom)
$\forall x_{14}, x_{15}, x_{16}$: $(x'_{14} >=' x_{15} \Rightarrow x'_{16} ==>' x'_{14} >=' x'_{16} ==>' x_{15})$ fof(sos_{11}, axiom)
$\forall a$: $a' +' \text{'1'} = \text{'1'}$ fof(sos_{12}, axiom)
$\forall a$: $a' ==>' \text{'1'}' ==>' \text{'1'} = a$ fof(sos_{13}, axiom)
$\forall x_{17}$: $\exists x_{18}$: $x_{18} = x'_{18} ==>' x_{17}$ fof(goals_{14}, conjecture)

LCL886+1.p An involutive pocrim that is not idempotent
$\forall a, b, c$: $a' +' b' +' c = a' +' b' +' c$ fof(sos_{01}, axiom)
$\forall a, b$: $a' +' b = b' +' a$ fof(sos_{02}, axiom)
$\forall a$: $a' +' \text{'0'} = a$ fof(sos_{03}, axiom)
$\forall a$: $a' >=' a$ fof(sos_{04}, axiom)
$\forall x_0, x_1, x_2$: $((x'_0 >=' x_1 \text{ and } x'_1 >=' x_2) \Rightarrow x'_0 >=' x_2)$ fof(sos_{05}, axiom)
$\forall x_3, x_4$: $((x'_3 >=' x_4 \text{ and } x'_4 >=' x_3) \Rightarrow x_3 = x_4)$ fof(sos_{06}, axiom)

$\forall x_5, x_6, x_7$: $(x_5' +' x_6' >=' x_7 \iff x_6' >=' x_5' ==>' x_7)$ fof(sos_{07}, axiom)
$\forall a$: $a' >=' \text{'0'}$ fof(sos_{08}, axiom)
$\forall x_8, x_9, x_{10}$: $(x_8' >=' x_9 \Rightarrow x_8' +' x_{10}' >=' x_9' +' x_{10})$ fof(sos_{09}, axiom)
$\forall x_{11}, x_{12}, x_{13}$: $(x_{11}' >=' x_{12} \Rightarrow x_{12}' ==>' x_{13}' >=' x_{11}' ==>' x_{13})$ fof(sos_{10}, axiom)
$\forall x_{14}, x_{15}, x_{16}$: $(x_{14}' >=' x_{15} \Rightarrow x_{16}' ==>' x_{14}' >=' x_{16}' ==>' x_{15})$ fof(sos_{11}, axiom)
$\forall a$: $a' +' \text{'1'} = \text{'1'}$ fof(sos_{12}, axiom)
$\forall a$: $a' ==>' \text{'1''} ==>' \text{'1'} = a$ fof(sos_{13}, axiom)
$\forall x_{17}$: $x_{17}' +' x_{17} = x_{17}$ fof(goals_{14}, conjecture)

LCL887+1.p An idempotent hoop that is not an involutive pocrim
$\forall a, b, c$: $a' +' b' +' c = a' +' b' +' c$ fof(sos_{01}, axiom)
$\forall a, b$: $a' +' b = b' +' a$ fof(sos_{02}, axiom)
$\forall a$: $a' +' \text{'0'} = a$ fof(sos_{03}, axiom)
$\forall a$: $a' >=' a$ fof(sos_{04}, axiom)
$\forall x_0, x_1, x_2$: $((x_0' >=' x_1 \text{ and } x_1' >=' x_2) \Rightarrow x_0' >=' x_2)$ fof(sos_{05}, axiom)
$\forall x_3, x_4$: $((x_3' >=' x_4 \text{ and } x_4' >=' x_3) \Rightarrow x_3 = x_4)$ fof(sos_{06}, axiom)
$\forall x_5, x_6, x_7$: $(x_5' +' x_6' >=' x_7 \iff x_6' >=' x_5' ==>' x_7)$ fof(sos_{07}, axiom)
$\forall a$: $a' >=' \text{'0'}$ fof(sos_{08}, axiom)
$\forall x_8, x_9, x_{10}$: $(x_8' >=' x_9 \Rightarrow x_8' +' x_{10}' >=' x_9' +' x_{10})$ fof(sos_{09}, axiom)
$\forall x_{11}, x_{12}, x_{13}$: $(x_{11}' >=' x_{12} \Rightarrow x_{12}' ==>' x_{13}' >=' x_{11}' ==>' x_{13})$ fof(sos_{10}, axiom)
$\forall x_{14}, x_{15}, x_{16}$: $(x_{14}' >=' x_{15} \Rightarrow x_{16}' ==>' x_{14}' >=' x_{16}' ==>' x_{15})$ fof(sos_{11}, axiom)
$\forall a$: $a' +' \text{'1'} = \text{'1'}$ fof(sos_{12}, axiom)
$\forall a$: $a' +' a = a$ fof(sos_{13}, axiom)
$\forall x_{17}$: $x_{17}' ==>' \text{'1''} ==>' \text{'1'} = x_{17}$ fof(goals_{14}, conjecture)

LCL888+1.p Halving is unique in a hoop
$\forall a, b, c$: $a' +' b' +' c = a' +' b' +' c$ fof(sos_{01}, axiom)
$\forall a, b$: $a' +' b = b' +' a$ fof(sos_{02}, axiom)
$\forall a$: $a' +' \text{'0'} = a$ fof(sos_{03}, axiom)
$\forall a$: $a' >=' a$ fof(sos_{04}, axiom)
$\forall x_0, x_1, x_2$: $((x_0' >=' x_1 \text{ and } x_1' >=' x_2) \Rightarrow x_0' >=' x_2)$ fof(sos_{05}, axiom)
$\forall x_3, x_4$: $((x_3' >=' x_4 \text{ and } x_4' >=' x_3) \Rightarrow x_3 = x_4)$ fof(sos_{06}, axiom)
$\forall x_5, x_6, x_7$: $(x_5' +' x_6' >=' x_7 \iff x_6' >=' x_5' ==>' x_7)$ fof(sos_{07}, axiom)
$\forall a$: $a' >=' \text{'0'}$ fof(sos_{08}, axiom)
$\forall x_8, x_9, x_{10}$: $(x_8' >=' x_9 \Rightarrow x_8' +' x_{10}' >=' x_9' +' x_{10})$ fof(sos_{09}, axiom)
$\forall x_{11}, x_{12}, x_{13}$: $(x_{11}' >=' x_{12} \Rightarrow x_{12}' ==>' x_{13}' >=' x_{11}' ==>' x_{13})$ fof(sos_{10}, axiom)
$\forall x_{14}, x_{15}, x_{16}$: $(x_{14}' >=' x_{15} \Rightarrow x_{16}' ==>' x_{14}' >=' x_{16}' ==>' x_{15})$ fof(sos_{11}, axiom)
$\forall a, b$: $a' +' a' ==>' b = b' +' b' ==>' a$ fof(sos_{12}, axiom)
$\forall x_{17}, x_{18}, x_{19}$: $((x_{17} = x_{17}' ==>' x_{18} \text{ and } x_{19} = x_{19}' ==>' x_{18}) \Rightarrow x_{17} = x_{19})$ fof(goals_{13}, conjecture)

LCL889+1.p Halving is unique in a hoop, rule for x >= a/2
$\forall a, b, c$: $a' +' b' +' c = a' +' b' +' c$ fof(sos_{01}, axiom)
$\forall a, b$: $a' +' b = b' +' a$ fof(sos_{02}, axiom)
$\forall a$: $a' +' \text{'0'} = a$ fof(sos_{03}, axiom)
$\forall a$: $a' >=' a$ fof(sos_{04}, axiom)
$\forall x_0, x_1, x_2$: $((x_0' >=' x_1 \text{ and } x_1' >=' x_2) \Rightarrow x_0' >=' x_2)$ fof(sos_{05}, axiom)
$\forall x_3, x_4$: $((x_3' >=' x_4 \text{ and } x_4' >=' x_3) \Rightarrow x_3 = x_4)$ fof(sos_{06}, axiom)
$\forall x_5, x_6, x_7$: $(x_5' +' x_6' >=' x_7 \iff x_6' >=' x_5' ==>' x_7)$ fof(sos_{07}, axiom)
$\forall a$: $a' >=' \text{'0'}$ fof(sos_{08}, axiom)
$\forall x_8, x_9, x_{10}$: $(x_8' >=' x_9 \Rightarrow x_8' +' x_{10}' >=' x_9' +' x_{10})$ fof(sos_{09}, axiom)
$\forall x_{11}, x_{12}, x_{13}$: $(x_{11}' >=' x_{12} \Rightarrow x_{12}' ==>' x_{13}' >=' x_{11}' ==>' x_{13})$ fof(sos_{10}, axiom)
$\forall x_{14}, x_{15}, x_{16}$: $(x_{14}' >=' x_{15} \Rightarrow x_{16}' ==>' x_{14}' >=' x_{16}' ==>' x_{15})$ fof(sos_{11}, axiom)
$\forall a, b$: $a' +' a' ==>' b = b' +' b' ==>' a$ fof(sos_{12}, axiom)
$\forall x_{17}, x_{18}, x_{19}$: $((x_{17}' >=' x_{17}' ==>' x_{18} \text{ and } x_{19} = x_{19}' ==>' x_{18}) \Rightarrow x_{17}' >=' x_{19})$ fof(goals_{13}, conjecture)

LCL890+1.p Halving is unique in a hoop, rule for a/2 >= x
$\forall a, b, c$: $a' +' b' +' c = a' +' b' +' c$ fof(sos_{01}, axiom)
$\forall a, b$: $a' +' b = b' +' a$ fof(sos_{02}, axiom)
$\forall a$: $a' +' \text{'0'} = a$ fof(sos_{03}, axiom)
$\forall a$: $a' +' \text{'1'} = \text{'1'}$ fof(sos_{04}, axiom)
$\forall a$: $a' >=' a$ fof(sos_{05}, axiom)

$\forall x_0, x_1, x_2\colon ((x'_0 >=' x_1 \text{ and } x'_1 >=' x_2) \Rightarrow x'_0 >=' x_2)$ fof(sos_{06}, axiom)

$\forall x_3, x_4\colon ((x'_3 >=' x_4 \text{ and } x'_4 >=' x_3) \Rightarrow x_3 = x_4)$ fof(sos_{07}, axiom)

$\forall x_5, x_6, x_7\colon (x'_5 +' x'_6 >=' x_7 \iff x'_6 >=' x'_5 ==>' x_7)$ fof(sos_{08}, axiom)

$\forall a\colon a' >=' \text{'0'}$ fof(sos_{09}, axiom)

$\forall x_8, x_9, x_{10}\colon (x'_8 >=' x_9 \Rightarrow x'_8 +' x'_{10} >=' x'_9 +' x_{10})$ fof(sos_{10}, axiom)

$\forall x_{11}, x_{12}, x_{13}\colon (x'_{11} >=' x_{12} \Rightarrow x'_{12} ==>' x'_{13} >=' x'_{11} ==>' x_{13})$ fof(sos_{11}, axiom)

$\forall x_{14}, x_{15}, x_{16}\colon (x'_{14} >=' x_{15} \Rightarrow x'_{16} ==>' x'_{14} >=' x'_{16} ==>' x_{15})$ fof(sos_{12}, axiom)

$\forall a, b\colon a' +' a' ==>' b = b' +' b' ==>' a$ fof(sos_{13}, axiom)

$\forall x_{17}, x_{18}, x_{19}\colon ((x'_{17} ==>' x'_{18} >=' x_{17} \text{ and } x_{19} = x'_{19} ==>' x_{18}) \Rightarrow x'_{19} >=' x_{17})$ fof(goals_{14}, conjecture)

LCL891+1.p Halving is unique in a hoop, rule for a/2 >= x

$\forall a, b, c\colon a' +' b' +' c = a' +' b' +' c$ fof(sos_{01}, axiom)

$\forall a, b\colon a' +' b = b' +' a$ fof(sos_{02}, axiom)

$\forall a\colon a' +' \text{'0'} = a$ fof(sos_{03}, axiom)

$\forall a\colon a' >=' a$ fof(sos_{04}, axiom)

$\forall x_0, x_1, x_2\colon ((x'_0 >=' x_1 \text{ and } x'_1 >=' x_2) \Rightarrow x'_0 >=' x_2)$ fof(sos_{05}, axiom)

$\forall x_3, x_4\colon ((x'_3 >=' x_4 \text{ and } x'_4 >=' x_3) \Rightarrow x_3 = x_4)$ fof(sos_{06}, axiom)

$\forall x_5, x_6, x_7\colon (x'_5 +' x'_6 >=' x_7 \iff x'_6 >=' x'_5 ==>' x_7)$ fof(sos_{07}, axiom)

$\forall a\colon a' >=' \text{'0'}$ fof(sos_{08}, axiom)

$\forall x_8, x_9, x_{10}\colon (x'_8 >=' x_9 \Rightarrow x'_8 +' x'_{10} >=' x'_9 +' x_{10})$ fof(sos_{09}, axiom)

$\forall x_{11}, x_{12}, x_{13}\colon (x'_{11} >=' x_{12} \Rightarrow x'_{12} ==>' x'_{13} >=' x'_{11} ==>' x_{13})$ fof(sos_{10}, axiom)

$\forall x_{14}, x_{15}, x_{16}\colon (x'_{14} >=' x_{15} \Rightarrow x'_{16} ==>' x'_{14} >=' x'_{16} ==>' x_{15})$ fof(sos_{11}, axiom)

$\forall a, b\colon a' +' a' ==>' b = b' +' b' ==>' a$ fof(sos_{12}, axiom)

$\forall x_{17}, x_{18}, x_{19}\colon ((x'_{17} ==>' x'_{18} >=' x_{17} \text{ and } x_{19} = x'_{19} ==>' x_{18}) \Rightarrow x'_{19} >=' x_{17})$ fof(goals_{13}, conjecture)

LCL892+1.p Halving is unique in a hoop, rule for a/2 >= x

$\forall a, b, c\colon a' +' b' +' c = a' +' b' +' c$ fof(sos_{01}, axiom)

$\forall a, b\colon a' +' b = b' +' a$ fof(sos_{02}, axiom)

$\forall a\colon a' +' \text{'0'} = a$ fof(sos_{03}, axiom)

$\forall a\colon a' >=' a$ fof(sos_{04}, axiom)

$\forall x_0, x_1, x_2\colon ((x'_0 >=' x_1 \text{ and } x'_1 >=' x_2) \Rightarrow x'_0 >=' x_2)$ fof(sos_{05}, axiom)

$\forall x_3, x_4\colon ((x'_3 >=' x_4 \text{ and } x'_4 >=' x_3) \Rightarrow x_3 = x_4)$ fof(sos_{06}, axiom)

$\forall x_5, x_6, x_7\colon (x'_5 +' x'_6 >=' x_7 \iff x'_6 >=' x'_5 ==>' x_7)$ fof(sos_{07}, axiom)

$\forall a\colon a' >=' \text{'0'}$ fof(sos_{08}, axiom)

$\forall x_8, x_9, x_{10}\colon (x'_8 >=' x_9 \Rightarrow x'_8 +' x'_{10} >=' x'_9 +' x_{10})$ fof(sos_{09}, axiom)

$\forall x_{11}, x_{12}, x_{13}\colon (x'_{11} >=' x_{12} \Rightarrow x'_{12} ==>' x'_{13} >=' x'_{11} ==>' x_{13})$ fof(sos_{10}, axiom)

$\forall x_{14}, x_{15}, x_{16}\colon (x'_{14} >=' x_{15} \Rightarrow x'_{16} ==>' x'_{14} >=' x'_{16} ==>' x_{15})$ fof(sos_{11}, axiom)

$\forall a, b\colon a' +' a' ==>' b = b' +' b' ==>' a$ fof(sos_{12}, axiom)

$\forall a\colon a' +' \text{'1'} = \text{'1'}$ fof(sos_{13}, axiom)

$\forall x_{17}, x_{18}, x_{19}\colon ((x'_{17} ==>' x'_{18} >=' x_{17} \text{ and } x_{19} = x'_{19} ==>' x_{18}) \Rightarrow x'_{19} >=' x_{17})$ fof(goals_{14}, conjecture)

LCL893+1.p In a coop, x/2 = x implies x = 0

$\forall a, b, c\colon a' +' b' +' c = a' +' b' +' c$ fof(sos_{01}, axiom)

$\forall a, b\colon a' +' b = b' +' a$ fof(sos_{02}, axiom)

$\forall a\colon a' +' \text{'0'} = a$ fof(sos_{03}, axiom)

$\forall a\colon a' +' \text{'1'} = \text{'1'}$ fof(sos_{04}, axiom)

$\forall a\colon a' >=' a$ fof(sos_{05}, axiom)

$\forall x_0, x_1, x_2\colon ((x'_0 >=' x_1 \text{ and } x'_1 >=' x_2) \Rightarrow x'_0 >=' x_2)$ fof(sos_{06}, axiom)

$\forall x_3, x_4\colon ((x'_3 >=' x_4 \text{ and } x'_4 >=' x_3) \Rightarrow x_3 = x_4)$ fof(sos_{07}, axiom)

$\forall x_5, x_6, x_7\colon (x'_5 +' x'_6 >=' x_7 \iff x'_6 >=' x'_5 ==>' x_7)$ fof(sos_{08}, axiom)

$\forall a\colon a' >=' \text{'0'}$ fof(sos_{09}, axiom)

$\forall x_8, x_9, x_{10}\colon (x'_8 >=' x_9 \Rightarrow x'_8 +' x'_{10} >=' x'_9 +' x_{10})$ fof(sos_{10}, axiom)

$\forall x_{11}, x_{12}, x_{13}\colon (x'_{11} >=' x_{12} \Rightarrow x'_{12} ==>' x'_{13} >=' x'_{11} ==>' x_{13})$ fof(sos_{11}, axiom)

$\forall x_{14}, x_{15}, x_{16}\colon (x'_{14} >=' x_{15} \Rightarrow x'_{16} ==>' x'_{14} >=' x'_{16} ==>' x_{15})$ fof(sos_{12}, axiom)

$\forall a, b\colon a' +' a' ==>' b = b' +' b' ==>' a$ fof(sos_{13}, axiom)

$\forall a\colon h(a) = h(a)' ==>' a$ fof(sos_{14}, axiom)

$\forall x_{17}\colon (h(x_{17}) = x_{17} \Rightarrow x_{17} = \text{'0'})$ fof(goals_{15}, conjecture)

LCL894+1.p Weak conjunction is lub in a hoop using horn axioms

$\forall a, b, c\colon a' +' b' +' c = a' +' b' +' c$ fof(sos_{01}, axiom)

$\forall a, b\colon a' +' b = b' +' a$ fof(sos_{02}, axiom)

$\forall a\colon a' +' \text{'0'} = a \qquad \text{fof}(\text{sos}_{03}, \text{axiom})$

$\forall a\colon a' >=' a \qquad \text{fof}(\text{sos}_{04}, \text{axiom})$

$\forall x_0, x_1, x_2\colon ((x_0' >=' x_1 \text{ and } x_1' >=' x_2) \Rightarrow x_0' >=' x_2) \qquad \text{fof}(\text{sos}_{05}, \text{axiom})$

$\forall x_3, x_4\colon ((x_3' >=' x_4 \text{ and } x_4' >=' x_3) \Rightarrow x_3 = x_4) \qquad \text{fof}(\text{sos}_{06}, \text{axiom})$

$\forall x_5, x_6, x_7\colon (x_5' +' x_6' >=' x_7 \iff x_6' >=' x_5' ==>' x_7) \qquad \text{fof}(\text{sos}_{07}, \text{axiom})$

$\forall a\colon a' >=' \text{'0'} \qquad \text{fof}(\text{sos}_{08}, \text{axiom})$

$\forall x_8, x_9, x_{10}\colon (x_8' >=' x_9 \Rightarrow x_8' +' x_{10}' >=' x_9' +' x_{10}) \qquad \text{fof}(\text{sos}_{09}, \text{axiom})$

$\forall x_{11}, x_{12}, x_{13}\colon (x_{11}' >=' x_{12} \Rightarrow x_{12}' ==>' x_{13}' >=' x_{11}' ==>' x_{13}) \qquad \text{fof}(\text{sos}_{10}, \text{axiom})$

$\forall x_{14}, x_{15}, x_{16}\colon (x_{14}' >=' x_{15} \Rightarrow x_{16}' ==>' x_{14}' >=' x_{16}' ==>' x_{15}) \qquad \text{fof}(\text{sos}_{11}, \text{axiom})$

$\forall a, b\colon a' +' a' ==>' b = b' +' b' ==>' a \qquad \text{fof}(\text{sos}_{12}, \text{axiom})$

$(c' >=' a \text{ and } c' >=' b) \iff c' >=' a' +' a' ==>' b \qquad \text{fof}(\text{goals}_{13}, \text{conjecture})$

LCL895+1.p Weak conjunction is lub in a hoop using equational axioms

$\forall a, b, c\colon a' +' b' +' c = a' +' b' +' c \qquad \text{fof}(\text{sos}_{01}, \text{axiom})$

$\forall a, b\colon a' +' b = b' +' a \qquad \text{fof}(\text{sos}_{02}, \text{axiom})$

$\forall a\colon a' +' \text{'0'} = a \qquad \text{fof}(\text{sos}_{03}, \text{axiom})$

$\forall a\colon a' ==>' a = \text{'0'} \qquad \text{fof}(\text{sos}_{04}, \text{axiom})$

$\forall a\colon a' ==>' \text{'0'} = \text{'0'} \qquad \text{fof}(\text{sos}_{05}, \text{axiom})$

$\forall a\colon \text{'0'}' ==>' a = a \qquad \text{fof}(\text{sos}_{06}, \text{axiom})$

$\forall a, b, c\colon a' +' b' ==>' c = a' ==>' b' ==>' c \qquad \text{fof}(\text{sos}_{07}, \text{axiom})$

$\forall a, b\colon a' +' a' ==>' b = b' +' b' ==>' a \qquad \text{fof}(\text{sos}_{08}, \text{axiom})$

$(c' ==>' a = \text{'0'} \text{ and } c' ==>' b = \text{'0'}) \iff c' ==>' a' +' a' ==>' b = \text{'0'} \qquad \text{fof}(\text{goals}_{09}, \text{conjecture})$

LCL896+1.p Associativity of weak conjunction implies commutativity

$\forall a, b, c\colon a' +' b' +' c = a' +' b' +' c \qquad \text{fof}(\text{sos}_{01}, \text{axiom})$

$\forall a, b\colon a' +' b = b' +' a \qquad \text{fof}(\text{sos}_{02}, \text{axiom})$

$\forall a\colon a' +' \text{'0'} = a \qquad \text{fof}(\text{sos}_{03}, \text{axiom})$

$\forall a\colon a' >=' a \qquad \text{fof}(\text{sos}_{04}, \text{axiom})$

$\forall x_0, x_1, x_2\colon ((x_0' >=' x_1 \text{ and } x_1' >=' x_2) \Rightarrow x_0' >=' x_2) \qquad \text{fof}(\text{sos}_{05}, \text{axiom})$

$\forall x_3, x_4\colon ((x_3' >=' x_4 \text{ and } x_4' >=' x_3) \Rightarrow x_3 = x_4) \qquad \text{fof}(\text{sos}_{06}, \text{axiom})$

$\forall x_5, x_6, x_7\colon (x_5' +' x_6' >=' x_7 \iff x_6' >=' x_5' ==>' x_7) \qquad \text{fof}(\text{sos}_{07}, \text{axiom})$

$\forall a\colon a' >=' \text{'0'} \qquad \text{fof}(\text{sos}_{08}, \text{axiom})$

$\forall x_8, x_9, x_{10}\colon (x_8' >=' x_9 \Rightarrow x_8' +' x_{10}' >=' x_9' +' x_{10}) \qquad \text{fof}(\text{sos}_{09}, \text{axiom})$

$\forall x_{11}, x_{12}, x_{13}\colon (x_{11}' >=' x_{12} \Rightarrow x_{12}' ==>' x_{13}' >=' x_{11}' ==>' x_{13}) \qquad \text{fof}(\text{sos}_{10}, \text{axiom})$

$\forall x_{14}, x_{15}, x_{16}\colon (x_{14}' >=' x_{15} \Rightarrow x_{16}' ==>' x_{14}' >=' x_{16}' ==>' x_{15}) \qquad \text{fof}(\text{sos}_{11}, \text{axiom})$

$\forall a, b, c\colon a' +' a' ==>' b' +' a' +' a' ==>' b' ==>' c = a' +' a' ==>' b' +' b' ==>' c \qquad \text{fof}(\text{sos}_{12}, \text{axiom})$

$\forall x_{17}, x_{18}\colon x_{17}' +' x_{17}' ==>' x_{18} = x_{18}' +' x_{18}' ==>' x_{17} \qquad \text{fof}(\text{goals}_{13}, \text{conjecture})$

LCL897+1.p Weak conjunction is associative in a hoop

$\forall a, b, c\colon a' +' b' +' c = a' +' b' +' c \qquad \text{fof}(\text{sos}_{01}, \text{axiom})$

$\forall a, b\colon a' +' b = b' +' a \qquad \text{fof}(\text{sos}_{02}, \text{axiom})$

$\forall a\colon a' +' \text{'0'} = a \qquad \text{fof}(\text{sos}_{03}, \text{axiom})$

$\forall a\colon a' ==>' a = \text{'0'} \qquad \text{fof}(\text{sos}_{04}, \text{axiom})$

$\forall a\colon a' ==>' \text{'0'} = \text{'0'} \qquad \text{fof}(\text{sos}_{05}, \text{axiom})$

$\forall a\colon \text{'0'}' ==>' a = a \qquad \text{fof}(\text{sos}_{06}, \text{axiom})$

$\forall a, b, c\colon a' +' b' ==>' c = a' ==>' b' ==>' c \qquad \text{fof}(\text{sos}_{07}, \text{axiom})$

$\forall a, b\colon a' +' a' ==>' b = b' +' b' ==>' a \qquad \text{fof}(\text{sos}_{08}, \text{axiom})$

$a' +' a' ==>' b' +' a' +' a' ==>' b' ==>' c = a' +' a' ==>' b' +' b' ==>' c \qquad \text{fof}(\text{goals}_{09}, \text{conjecture})$

LCL898+1.p Strong disjunction is commutative in an involutive hoop

$\forall a, b, c\colon a' +' b' +' c = a' +' b' +' c \qquad \text{fof}(\text{sos}_{01}, \text{axiom})$

$\forall a, b\colon a' +' b = b' +' a \qquad \text{fof}(\text{sos}_{02}, \text{axiom})$

$\forall a\colon a' +' \text{'0'} = a \qquad \text{fof}(\text{sos}_{03}, \text{axiom})$

$\forall a\colon a' >=' a \qquad \text{fof}(\text{sos}_{04}, \text{axiom})$

$\forall x_0, x_1, x_2\colon ((x_0' >=' x_1 \text{ and } x_1' >=' x_2) \Rightarrow x_0' >=' x_2) \qquad \text{fof}(\text{sos}_{05}, \text{axiom})$

$\forall x_3, x_4\colon ((x_3' >=' x_4 \text{ and } x_4' >=' x_3) \Rightarrow x_3 = x_4) \qquad \text{fof}(\text{sos}_{06}, \text{axiom})$

$\forall x_5, x_6, x_7\colon (x_5' +' x_6' >=' x_7 \iff x_6' >=' x_5' ==>' x_7) \qquad \text{fof}(\text{sos}_{07}, \text{axiom})$

$\forall a\colon a' >=' \text{'0'} \qquad \text{fof}(\text{sos}_{08}, \text{axiom})$

$\forall x_8, x_9, x_{10}\colon (x_8' >=' x_9 \Rightarrow x_8' +' x_{10}' >=' x_9' +' x_{10}) \qquad \text{fof}(\text{sos}_{09}, \text{axiom})$

$\forall x_{11}, x_{12}, x_{13}\colon (x_{11}' >=' x_{12} \Rightarrow x_{12}' ==>' x_{13}' >=' x_{11}' ==>' x_{13}) \qquad \text{fof}(\text{sos}_{10}, \text{axiom})$

$\forall x_{14}, x_{15}, x_{16}\colon (x_{14}' >=' x_{15} \Rightarrow x_{16}' ==>' x_{14}' >=' x_{16}' ==>' x_{15}) \qquad \text{fof}(\text{sos}_{11}, \text{axiom})$

$\forall a, b$: $a' +' a' ==>' b = b' +' b' ==>' a$ fof(sos_{12}, axiom)
$\forall a$: $a' ==>'$ '1'' $==>'$ '1' $= a$ fof(sos_{13}, axiom)
$\forall a$: $a' +'$ '1' = '1' fof(sos_{14}, axiom)
$\forall x_{17}, x_{18}$: $x'_{17} ==>' x'_{18} ==>' x_{18} = x'_{18} ==>' x'_{17} ==>' x_{17}$ fof(goals_{15}, conjecture)

LCL899+1.p A bounded pocrim property
A bounded pocrim with commutative strong disjunction is involutive.
$\forall a, b, c$: $a' +' b' +' c = a' +' b' +' c$ fof(sos_{01}, axiom)
$\forall a, b$: $a' +' b = b' +' a$ fof(sos_{02}, axiom)
$\forall a$: $a' +'$ '0' $= a$ fof(sos_{03}, axiom)
$\forall a$: $a' >=' a$ fof(sos_{04}, axiom)
$\forall x_0, x_1, x_2$: $((x'_0 >=' x_1$ and $x'_1 >=' x_2) \Rightarrow x'_0 >=' x_2)$ fof(sos_{05}, axiom)
$\forall x_3, x_4$: $((x'_3 >=' x_4$ and $x'_4 >=' x_3) \Rightarrow x_3 = x_4)$ fof(sos_{06}, axiom)
$\forall x_5, x_6, x_7$: $(x'_5 +' x'_6 >=' x_7 \iff x'_6 >=' x'_5 ==>' x_7)$ fof(sos_{07}, axiom)
$\forall a$: $a' >='$ '0' fof(sos_{08}, axiom)
$\forall x_8, x_9, x_{10}$: $(x'_8 >=' x_9 \Rightarrow x'_8 +' x'_{10} >=' x'_9 +' x_{10})$ fof(sos_{09}, axiom)
$\forall x_{11}, x_{12}, x_{13}$: $(x'_{11} >=' x_{12} \Rightarrow x'_{12} ==>' x'_{13} >=' x'_{11} ==>' x_{13})$ fof(sos_{10}, axiom)
$\forall x_{14}, x_{15}, x_{16}$: $(x'_{14} >=' x_{15} \Rightarrow x'_{16} ==>' x'_{14} >=' x'_{16} ==>' x_{15})$ fof(sos_{11}, axiom)
$\forall a$: $a' +'$ '1' = '1' fof(sos_{12}, axiom)
$\forall a, b$: $a' ==>' b' ==>' b = b' ==>' a' ==>' a$ fof(sos_{13}, axiom)
$\forall x_{17}$: $x'_{17} ==>'$ '1'' $==>'$ '1' $= x_{17}$ fof(goals_{14}, conjecture)

LCL900+1.p A bounded pocrim with commutative strong disjunction is a hoop
$\forall a, b, c$: $a' +' b' +' c = a' +' b' +' c$ fof(sos_{01}, axiom)
$\forall a, b$: $a' +' b = b' +' a$ fof(sos_{02}, axiom)
$\forall a$: $a' +'$ '0' $= a$ fof(sos_{03}, axiom)
$\forall a$: $a' >=' a$ fof(sos_{04}, axiom)
$\forall x_0, x_1, x_2$: $((x'_0 >=' x_1$ and $x'_1 >=' x_2) \Rightarrow x'_0 >=' x_2)$ fof(sos_{05}, axiom)
$\forall x_3, x_4$: $((x'_3 >=' x_4$ and $x'_4 >=' x_3) \Rightarrow x_3 = x_4)$ fof(sos_{06}, axiom)
$\forall x_5, x_6, x_7$: $(x'_5 +' x'_6 >=' x_7 \iff x'_6 >=' x'_5 ==>' x_7)$ fof(sos_{07}, axiom)
$\forall a$: $a' >='$ '0' fof(sos_{08}, axiom)
$\forall x_8, x_9, x_{10}$: $(x'_8 >=' x_9 \Rightarrow x'_8 +' x'_{10} >=' x'_9 +' x_{10})$ fof(sos_{09}, axiom)
$\forall x_{11}, x_{12}, x_{13}$: $(x'_{11} >=' x_{12} \Rightarrow x'_{12} ==>' x'_{13} >=' x'_{11} ==>' x_{13})$ fof(sos_{10}, axiom)
$\forall x_{14}, x_{15}, x_{16}$: $(x'_{14} >=' x_{15} \Rightarrow x'_{16} ==>' x'_{14} >=' x'_{16} ==>' x_{15})$ fof(sos_{11}, axiom)
$\forall a$: $a' +'$ '1' = '1' fof(sos_{12}, axiom)
$\forall a, b$: $a' ==>' b' ==>' b = b' ==>' a' ==>' a$ fof(sos_{13}, axiom)
$\forall x_{17}, x_{18}$: $x'_{17} +' x'_{17} ==>' x_{18} = x'_{18} +' x'_{18} ==>' x_{17}$ fof(goals_{14}, conjecture)

LCL901+1.p An idempotent pocrim property
An idempotent pocrim with commutative strong disjunction is boolean.
$\forall a, b, c$: $a' +' b' +' c = a' +' b' +' c$ fof(sos_{01}, axiom)
$\forall a, b$: $a' +' b = b' +' a$ fof(sos_{02}, axiom)
$\forall a$: $a' +'$ '0' $= a$ fof(sos_{03}, axiom)
$\forall a$: $a' >=' a$ fof(sos_{04}, axiom)
$\forall x_0, x_1, x_2$: $((x'_0 >=' x_1$ and $x'_1 >=' x_2) \Rightarrow x'_0 >=' x_2)$ fof(sos_{05}, axiom)
$\forall x_3, x_4$: $((x'_3 >=' x_4$ and $x'_4 >=' x_3) \Rightarrow x_3 = x_4)$ fof(sos_{06}, axiom)
$\forall x_5, x_6, x_7$: $(x'_5 +' x'_6 >=' x_7 \iff x'_6 >=' x'_5 ==>' x_7)$ fof(sos_{07}, axiom)
$\forall a$: $a' >='$ '0' fof(sos_{08}, axiom)
$\forall x_8, x_9, x_{10}$: $(x'_8 >=' x_9 \Rightarrow x'_8 +' x'_{10} >=' x'_9 +' x_{10})$ fof(sos_{09}, axiom)
$\forall x_{11}, x_{12}, x_{13}$: $(x'_{11} >=' x_{12} \Rightarrow x'_{12} ==>' x'_{13} >=' x'_{11} ==>' x_{13})$ fof(sos_{10}, axiom)
$\forall x_{14}, x_{15}, x_{16}$: $(x'_{14} >=' x_{15} \Rightarrow x'_{16} ==>' x'_{14} >=' x'_{16} ==>' x_{15})$ fof(sos_{11}, axiom)
$\forall a$: $a' +'$ '1' = '1' fof(sos_{12}, axiom)
$\forall a, b$: $a' ==>' b' ==>' b = b' ==>' a' ==>' a$ fof(sos_{13}, axiom)
$\forall a$: $a' +' a = a$ fof(sos_{14}, axiom)
$\forall x_{17}$: $x'_{17} ==>'$ '1'' $==>' x'_{17} ==>' x_{17} =$ '0' fof(goals_{15}, conjecture)

LCL902+1.p A boolean pocrim is involutive
$\forall a, b, c$: $a' +' b' +' c = a' +' b' +' c$ fof(sos_{01}, axiom)
$\forall a, b$: $a' +' b = b' +' a$ fof(sos_{02}, axiom)
$\forall a$: $a' +'$ '0' $= a$ fof(sos_{03}, axiom)
$\forall a$: $a' >=' a$ fof(sos_{04}, axiom)

$\forall x_0, x_1, x_2$: $((x_0' >=' x_1 \text{ and } x_1' >=' x_2) \Rightarrow x_0' >=' x_2)$ fof(sos_{05}, axiom)
$\forall x_3, x_4$: $((x_3' >=' x_4 \text{ and } x_4' >=' x_3) \Rightarrow x_3 = x_4)$ fof(sos_{06}, axiom)
$\forall x_5, x_6, x_7$: $(x_5' +' x_6' >=' x_7 \iff x_6' >=' x_5' ==>' x_7)$ fof(sos_{07}, axiom)
$\forall a$: $a' >='$ '0' fof(sos_{08}, axiom)
$\forall x_8, x_9, x_{10}$: $(x_8' >=' x_9 \Rightarrow x_8' +' x_{10}' >=' x_9' +' x_{10})$ fof(sos_{09}, axiom)
$\forall x_{11}, x_{12}, x_{13}$: $(x_{11}' >=' x_{12} \Rightarrow x_{12}' ==>' x_{13}' >=' x_{11}' ==>' x_{13})$ fof(sos_{10}, axiom)
$\forall x_{14}, x_{15}, x_{16}$: $(x_{14}' >=' x_{15} \Rightarrow x_{16}' ==>' x_{14}' >=' x_{16}' ==>' x_{15})$ fof(sos_{11}, axiom)
$\forall a$: $a' +'$ '1' = '1' fof(sos_{12}, axiom)
$\forall a$: $a' ==>'$ '1'$' ==>' a' ==>' a =$ '0' fof(sos_{13}, axiom)
$\forall x_{17}$: $x_{17}' ==>'$ '1'$' ==>'$ '1' $= x_{17}$ fof(goals_{14}, conjecture)

LCL903+1.p A boolean pocrim is idempotent
$\forall a, b, c$: $a' +' b' +' c = a' +' b' +' c$ fof(sos_{01}, axiom)
$\forall a, b$: $a' +' b = b' +' a$ fof(sos_{02}, axiom)
$\forall a$: $a' +'$ '0' $= a$ fof(sos_{03}, axiom)
$\forall a$: $a' >=' a$ fof(sos_{04}, axiom)
$\forall x_0, x_1, x_2$: $((x_0' >=' x_1 \text{ and } x_1' >=' x_2) \Rightarrow x_0' >=' x_2)$ fof(sos_{05}, axiom)
$\forall x_3, x_4$: $((x_3' >=' x_4 \text{ and } x_4' >=' x_3) \Rightarrow x_3 = x_4)$ fof(sos_{06}, axiom)
$\forall x_5, x_6, x_7$: $(x_5' +' x_6' >=' x_7 \iff x_6' >=' x_5' ==>' x_7)$ fof(sos_{07}, axiom)
$\forall a$: $a' >='$ '0' fof(sos_{08}, axiom)
$\forall x_8, x_9, x_{10}$: $(x_8' >=' x_9 \Rightarrow x_8' +' x_{10}' >=' x_9' +' x_{10})$ fof(sos_{09}, axiom)
$\forall x_{11}, x_{12}, x_{13}$: $(x_{11}' >=' x_{12} \Rightarrow x_{12}' ==>' x_{13}' >=' x_{11}' ==>' x_{13})$ fof(sos_{10}, axiom)
$\forall x_{14}, x_{15}, x_{16}$: $(x_{14}' >=' x_{15} \Rightarrow x_{16}' ==>' x_{14}' >=' x_{16}' ==>' x_{15})$ fof(sos_{11}, axiom)
$\forall a$: $a' +'$ '1' = '1' fof(sos_{12}, axiom)
$\forall a$: $a' ==>'$ '1'$' ==>' a' ==>' a =$ '0' fof(sos_{13}, axiom)
$\forall x_{17}$: $x_{17}' +' x_{17} = x_{17}$ fof(goals_{14}, conjecture)

LCL904^1.p Axioms for Modal logic S4 under cumulative domains
include('Axioms/LCL013^5.ax')
include('Axioms/LCL015^0.ax')
include('Axioms/LCL015^1.ax')

LCL905-1.p Alternative Wajsberg algebra
include('Axioms/LCL002-0.ax')
include('Axioms/LCL002-0.ax')

LCL906-1.p Lattice theory (equality) axioms
include('Axioms/LAT001-0.ax')
include('Axioms/LAT001-1.ax')
include('Axioms/LAT001-2.ax')
include('Axioms/LAT001-3.ax')

LCL907+1.p Hilbert's axiomatization of propositional logic
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+2.ax')

LCL908+1.p Lukasiewicz's axiomatization of propositional logic
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+3.ax')

LCL909+1.p Principia's axiomatization of propositional logic
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+4.ax')

LCL910+1.p Rosser's axiomatization of propositional logic
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+5.ax')

LCL911+1.p KM5 axiomatization of S5 based on Hilbert's PC
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')

include('Axioms/LCL006+2.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+2.ax')

LCL912+1.p KM4B axiomatization of S5 based on Hilbert's PC
include('Axioms/LCL006+0.ax')
include('Axioms/LCL006+1.ax')
include('Axioms/LCL006+2.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+3.ax')

LCL913+1.p Axiomatization of S1-0
include('Axioms/LCL006+1.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+4.ax')

LCL914+1.p M6S3M9B axiomatization of S5 based on S1-0
include('Axioms/LCL006+1.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+4.ax')
include('Axioms/LCL007+5.ax')

LCL915+1.p M10 axiomatization of S5 based on S1-0
include('Axioms/LCL006+1.ax')
include('Axioms/LCL007+0.ax')
include('Axioms/LCL007+1.ax')
include('Axioms/LCL007+4.ax')
include('Axioms/LCL007+6.ax')

LCL916^1.p Multi-Modal Logic
include('Axioms/LCL008ˆ0.ax')

LCL917^1.p Translating constructive S4 (CS4) to bimodal classical S4 (BS4)
include('Axioms/LCL008ˆ0.ax')
include('Axioms/LCL009ˆ0.ax')

LCL918^1.p Modal logic K
include('Axioms/LCL013ˆ0.ax')
include('Axioms/LCL013ˆ1.ax')

LCL919^1.p Modal logic D
include('Axioms/LCL013ˆ0.ax')
include('Axioms/LCL013ˆ2.ax')

LCL920^1.p Modal logic M
include('Axioms/LCL013ˆ0.ax')
include('Axioms/LCL013ˆ3.ax')

LCL921^1.p Modal logic B
include('Axioms/LCL013ˆ0.ax')
include('Axioms/LCL013ˆ4.ax')

LCL922^1.p Modal logic S4
include('Axioms/LCL013ˆ0.ax')
include('Axioms/LCL013ˆ5.ax')

LCL923^1.p Modal logic S5
include('Axioms/LCL013ˆ0.ax')
include('Axioms/LCL013ˆ6.ax')

LCL924^1.p Region Connection Calculus
include('Axioms/LCL013ˆ0.ax')
include('Axioms/LCL014ˆ0.ax')

LCL925∧1.p Embedding of quantified multimodal logic in simple type theory
include('Axioms/LCL015^0.ax')
include('Axioms/LCL015^1.ax')
include('Axioms/LCL013^5.ax')

LCL926-1.p IO in TW+ [AxL,AxTO]
$(p(i(a,b)) \text{ and } p(a)) \Rightarrow p(b)$ cnf(modus_ponens, axiom)
$p(i(i(a,b),i(i(b,c),i(a,c))))$ cnf(axBp, axiom)
$p(i(i(i(x,y),y),i(i(y,x),x)))$ cnf(axL, axiom)
$p(i(i(i(x,y),i(y,x)),i(y,x)))$ cnf(axTO, axiom)
$p(i(x,\text{or}(x,y)))$ cnf(axorI$_1$, axiom)
$p(i(y,\text{or}(x,y)))$ cnf(axorI$_2$, axiom)
$\neg p(i(i(i(x,y),y),\text{or}(x,y)))$ cnf(io, negated_conjecture)

LCL927-1.p AxK and AxC in TW+ [AxL,AxTO] + (Resid)
$(p(i(a,b)) \text{ and } p(a)) \Rightarrow p(b)$ cnf(modus_ponens, axiom)
$p(i(f(a,b),c)) \Rightarrow p(i(a,i(b,c)))$ cnf(resid$_1$, axiom)
$p(i(a,i(b,c))) \Rightarrow p(i(f(a,b),c))$ cnf(resid$_2$, axiom)
$p(i(i(a,b),i(i(b,c),i(a,c))))$ cnf(axBp, axiom)
$p(i(i(i(x,y),y),i(i(y,x),x)))$ cnf(axL, axiom)
$p(i(i(i(x,y),i(y,x)),i(y,x)))$ cnf(axTO, axiom)
$p(i(c_1,i(c_2,c_1))) \Rightarrow \neg p(i(i(a,i(b,c)),i(b,i(a,c))))$ cnf(axK_axC, negated_conjecture)

LCL928-1.p AxTO in BCK→ [AxL] + (Resid)
$(p(i(a,b)) \text{ and } p(a)) \Rightarrow p(b)$ cnf(modus_ponens, axiom)
$p(i(f(a,b),c)) \Rightarrow p(i(a,i(b,c)))$ cnf(resid$_1$, axiom)
$p(i(a,i(b,c))) \Rightarrow p(i(f(a,b),c))$ cnf(resid$_2$, axiom)
$p(i(i(a,b),i(i(b,c),i(a,c))))$ cnf(axBp, axiom)
$p(i(i(i(a,b),b),i(i(b,a),a)))$ cnf(axL, axiom)
$p(i(i(a,i(b,c)),i(b,i(a,c))))$ cnf(axC, axiom)
$\neg p(i(i(i(c_1,c_2),i(c_2,c_1)),i(c_2,c_1)))$ cnf(axTO, negated_conjecture)

LCL929-1.p AxK in TW→ [AxL] + (Resid)
$(p(i(a,b)) \text{ and } p(a)) \Rightarrow p(b)$ cnf(modus_ponens, axiom)
$p(i(f(a,b),c)) \Rightarrow p(i(a,i(b,c)))$ cnf(resid$_1$, axiom)
$p(i(a,i(b,c))) \Rightarrow p(i(f(a,b),c))$ cnf(resid$_2$, axiom)
$p(i(a,a))$ cnf(axI, axiom)
$p(i(i(a,b),i(i(b,c),i(a,c))))$ cnf(axBp, axiom)
$p(i(i(i(a,b),b),i(i(b,a),a)))$ cnf(axL, axiom)
$\neg p(i(a,i(b,a)))$ cnf(axK, negated_conjecture)

LCL930∧1.p Embedding of second order modal logic S5 with universal access
include('Axioms/LCL017^0.ax')