GEG axioms
GEG problems

GEG002∧1.p Catalunya and Paris, and Spain and Paris, are disconnected
The assumptions express that Catalunya is a border region of Spain, Spain and France are two different countries sharing a common border, and Paris is a proper part of France. The conjecture is that Catalunya and Paris are disconnected as well as Spain and Paris.
include('Axioms/LCL014ˆ0.ax')
catalunya: reg   thf(catalunya, type)
france: reg   thf(france, type)
spain: reg   thf(spain, type)
paris: reg   thf(paris, type)
tpp@catalunya@spain   thf(ax_1, axiom)
ec@spain@france   thf(ax_2, axiom)
ntpp@paris@france   thf(ax_3, axiom)
dc@catalunya@paris and dc@spain@paris   thf(con, conjecture)

GEG003∧1.p Bob knows Catalunya and Paris and Spain and Paris are disconnected
We here express that some spatial relations about Catalunya, France, Spain, and Paris are commonly known (modality box_fool), while others are known only to person Bob (modality box_bob). We prove that Bob knows that Catalunya and Paris and Spain and Paris are disconnected.
include('Axioms/LCL013ˆ0.ax')
include('Axioms/LCL014ˆ0.ax')
catalunya: reg   thf(catalunya, type)
france: reg   thf(france, type)
spain: reg   thf(spain, type)
paris: reg   thf(paris, type)
a: $i → $i → $o   thf(*a*, type)
fool: $i → $i → $o   thf(fool, type)
mvalid@(mforall_prop@λphi: $i → $o: (mimplies@(mbox@fool@phi)@(mbox@*a*@phi)))   thf(i_axiom_for_fool_a, axiom)
mvalid@(mbox@*a*@λ*x*: $i: (tpp@catalunya@spain))   thf(ax_1, axiom)
mvalid@(mbox@fool@λ*x*: $i: (ec@spain@france))   thf(ax_2, axiom)
mvalid@(mbox@*a*@λ*x*: $i: (ntpp@paris@france))   thf(ax_3, axiom)
mvalid@(mbox@*a*@λ*x*: $i: (dc@catalunya@paris and dc@spain@paris))   thf(con, conjecture)

GEG004∧1.p Is it commonly known that places are disconnected?
We here express that some spatial relations about Catalunya, France, Spain, and Paris are commonly known (modality box_fool), while others are known only to person Bob (modality box_bob). We ask whether it is commonly known that Catalunya and Paris and Spain and Paris are disconnected. (This is not the case).
include('Axioms/LCL013ˆ0.ax')
include('Axioms/LCL014ˆ0.ax')
catalunya: reg   thf(catalunya, type)
france: reg   thf(france, type)
spain: reg   thf(spain, type)
paris: reg   thf(paris, type)
a: $i → $i → $o   thf(*a*, type)
fool: $i → $i → $o   thf(fool, type)
mvalid@(mforall_prop@λ*a*: $i → $o: (mimplies@(mbox@fool@*a*)@*a*))   thf(t_axiom_for_fool, axiom)
mvalid@(mforall_prop@λ*a*: $i → $o: (mimplies@(mbox@fool@*a*)@(mbox@fool@(mbox@fool@*a*))))   thf(k_axiom_for_fool, a
mvalid@(mforall_prop@λphi: $i → $o: (mimplies@(mbox@fool@phi)@(mbox@*a*@phi)))   thf(i_axiom_for_fool_a, axiom)
mvalid@(mbox@*a*@λ*x*: $i: (tpp@catalunya@spain))   thf(ax_1, axiom)
mvalid@(mbox@fool@λ*x*: $i: (ec@spain@france))   thf(ax_2, axiom)
mvalid@(mbox@*a*@λ*x*: $i: (ntpp@paris@france))   thf(ax_3, axiom)
mvalid@(mbox@fool@λ*x*: $i: (dc@catalunya@paris and dc@spain@paris))   thf(con, conjecture)

GEG005∧1.p Catalunya is not part of Paris
include('Axioms/LCL013ˆ0.ax')
include('Axioms/LCL014ˆ0.ax')
catalunya: reg   thf(catalunya, type)

france: reg   thf(france, type)
spain: reg   thf(spain, type)
paris: reg   thf(paris, type)
a: $i → $i → $o   thf(*a*, type)
fool: $i → $i → $o   thf(fool, type)
mvalid@(mforall_prop@λ*a*: $i → $o: (mimplies@(mbox@fool@*a*)@*a*))   thf(t_axiom_for_fool, axiom)
mvalid@(mforall_prop@λ*a*: $i → $o: (mimplies@(mbox@fool@*a*)@(mbox@fool@(mbox@fool@*a*))))   thf(k_axiom_for_fool, a
mvalid@(mforall_prop@λphi: $i → $o: (mimplies@(mbox@fool@phi)@(mbox@*a*@phi)))   thf(i_axiom_for_fool_a, axiom)
mvalid@(mbox@*a*@λ*x*: $i: (tpp@catalunya@spain))   thf(ax₁, axiom)
mvalid@(mbox@fool@λ*x*: $i: (ec@spain@france))   thf(ax₂, axiom)
mvalid@(mbox@*a*@λ*x*: $i: (ntpp@paris@france))   thf(ax₃, axiom)
mvalid@(mbox@*a*@λ*x*: $i: ¬ po@catalunya@paris)   thf(con, conjecture)

GEG006^1.p Catalunya is in Spain and Paris is in France
include('Axioms/LCL013^0.ax')
include('Axioms/LCL014^0.ax')
catalunya: reg   thf(catalunya, type)
france: reg   thf(france, type)
spain: reg   thf(spain, type)
paris: reg   thf(paris, type)
a: $i → $i → $o   thf(*a*, type)
fool: $i → $i → $o   thf(fool, type)
mvalid@(mforall_prop@λ*a*: $i → $o: (mimplies@(mbox@fool@*a*)@*a*))   thf(t_axiom_for_fool, axiom)
mvalid@(mforall_prop@λ*a*: $i → $o: (mimplies@(mbox@fool@*a*)@(mbox@fool@(mbox@fool@*a*))))   thf(k_axiom_for_fool, a
mvalid@(mforall_prop@λphi: $i → $o: (mimplies@(mbox@fool@phi)@(mbox@*a*@phi)))   thf(i_axiom_for_fool_a, axiom)
mvalid@(mbox@*a*@λ*x*: $i: (tpp@catalunya@spain))   thf(ax₁, axiom)
mvalid@(mbox@fool@λ*x*: $i: (ec@spain@france))   thf(ax₂, axiom)
mvalid@(mbox@*a*@λ*x*: $i: (ntpp@paris@france))   thf(ax₃, axiom)
mvalid@(mbox@*a*@λ*x*: $i: (*o*@catalunya@spain and *o*@france@paris))   thf(con, conjecture)

GEG007^1.p Catalunya is not Paris
include('Axioms/LCL013^0.ax')
include('Axioms/LCL014^0.ax')
catalunya: reg   thf(catalunya, type)
france: reg   thf(france, type)
spain: reg   thf(spain, type)
paris: reg   thf(paris, type)
a: $i → $i → $o   thf(*a*, type)
fool: $i → $i → $o   thf(fool, type)
mvalid@(mforall_prop@λ*a*: $i → $o: (mimplies@(mbox@fool@*a*)@*a*))   thf(t_axiom_for_fool, axiom)
mvalid@(mforall_prop@λ*a*: $i → $o: (mimplies@(mbox@fool@*a*)@(mbox@fool@(mbox@fool@*a*))))   thf(k_axiom_for_fool, a
mvalid@(mforall_prop@λphi: $i → $o: (mimplies@(mbox@fool@phi)@(mbox@*a*@phi)))   thf(i_axiom_for_fool_a, axiom)
mvalid@(mbox@*a*@λ*x*: $i: (tpp@catalunya@spain))   thf(ax₁, axiom)
mvalid@(mbox@fool@λ*x*: $i: (ec@spain@france))   thf(ax₂, axiom)
mvalid@(mbox@*a*@λ*x*: $i: (ntpp@paris@france))   thf(ax₃, axiom)
mvalid@(mbox@*a*@λ*x*: $i: ¬ eq@catalunya@paris)   thf(con, conjecture)

GEG008^1.p If Catalunya is Paris then France is Spain
include('Axioms/LCL013^0.ax')
include('Axioms/LCL014^0.ax')
catalunya: reg   thf(catalunya, type)
france: reg   thf(france, type)
spain: reg   thf(spain, type)
paris: reg   thf(paris, type)
a: $i → $i → $o   thf(*a*, type)
fool: $i → $i → $o   thf(fool, type)
mvalid@(mforall_prop@λ*a*: $i → $o: (mimplies@(mbox@fool@*a*)@*a*))   thf(t_axiom_for_fool, axiom)
mvalid@(mforall_prop@λ*a*: $i → $o: (mimplies@(mbox@fool@*a*)@(mbox@fool@(mbox@fool@*a*))))   thf(k_axiom_for_fool, a
mvalid@(mforall_prop@λphi: $i → $o: (mimplies@(mbox@fool@phi)@(mbox@*a*@phi)))   thf(i_axiom_for_fool_a, axiom)
mvalid@(mbox@*a*@λ*x*: $i: (tpp@catalunya@spain))   thf(ax₁, axiom)
mvalid@(mbox@fool@λ*x*: $i: (ec@spain@france))   thf(ax₂, axiom)

mvalid@(mbox@a@λx: \$i: (ntpp@paris@france))   thf(ax_3, axiom)
mvalid@(mbox@a@λx: \$i: ((eq@catalunya@paris) ⇒ (o@france@spain)))   thf(con, conjecture)

GEG009∧1.p Something about Paris and Spain
include('Axioms/LCL013ˆ0.ax')
include('Axioms/LCL014ˆ0.ax')
catalunya: reg   thf(catalunya, type)
france: reg   thf(france, type)
spain: reg   thf(spain, type)
paris: reg   thf(paris, type)
a: \$i → \$i → \$o   thf(a, type)
fool: \$i → \$i → \$o   thf(fool, type)
mvalid@(mforall_prop@λa: \$i → \$o: (mimplies@(mbox@fool@a)@a))   thf(t_axiom_for_fool, axiom)
mvalid@(mforall_prop@λa: \$i → \$o: (mimplies@(mbox@fool@a)@(mbox@fool@(mbox@fool@a))))   thf(k_axiom_for_fool, a
mvalid@(mforall_prop@λphi: \$i → \$o: (mimplies@(mbox@fool@phi)@(mbox@a@phi)))   thf(i_axiom_for_fool_a, axiom)
mvalid@(mbox@a@λx: \$i: (tpp@catalunya@spain))   thf(ax_1, axiom)
mvalid@(mbox@fool@λx: \$i: (ec@spain@france))   thf(ax_2, axiom)
mvalid@(mbox@a@λx: \$i: (ntpp@paris@france))   thf(ax_3, axiom)
mvalid@(mbox@a@λx: \$i: ∃z: reg: (¬ o@z@paris and ¬ eq@z@spain))   thf(con, conjecture)

GEG010∧1.p Catalunya and Paris are not in the same place
include('Axioms/LCL013ˆ0.ax')
include('Axioms/LCL014ˆ0.ax')
catalunya: reg   thf(catalunya, type)
france: reg   thf(france, type)
spain: reg   thf(spain, type)
paris: reg   thf(paris, type)
a: \$i → \$i → \$o   thf(a, type)
fool: \$i → \$i → \$o   thf(fool, type)
mvalid@(mforall_prop@λa: \$i → \$o: (mimplies@(mbox@fool@a)@a))   thf(t_axiom_for_fool, axiom)
mvalid@(mforall_prop@λa: \$i → \$o: (mimplies@(mbox@fool@a)@(mbox@fool@(mbox@fool@a))))   thf(k_axiom_for_fool, a
mvalid@(mforall_prop@λphi: \$i → \$o: (mimplies@(mbox@fool@phi)@(mbox@a@phi)))   thf(i_axiom_for_fool_a, axiom)
mvalid@(mbox@a@λx: \$i: (tpp@catalunya@spain))   thf(ax_1, axiom)
mvalid@(mbox@fool@λx: \$i: (ec@spain@france))   thf(ax_2, axiom)
mvalid@(mbox@a@λx: \$i: (ntpp@paris@france))   thf(ax_3, axiom)
mvalid@(mbox@a@λx: \$i: ∀z: reg: ((p@z@catalunya) ⇒ ¬ p@z@paris))   thf(con, conjecture)

GEG011∧1.p Something about France, Spain, Paris, Catalunya
include('Axioms/LCL013ˆ0.ax')
include('Axioms/LCL014ˆ0.ax')
catalunya: reg   thf(catalunya, type)
france: reg   thf(france, type)
spain: reg   thf(spain, type)
paris: reg   thf(paris, type)
a: \$i → \$i → \$o   thf(a, type)
fool: \$i → \$i → \$o   thf(fool, type)
mvalid@(mforall_prop@λa: \$i → \$o: (mimplies@(mbox@fool@a)@a))   thf(t_axiom_for_fool, axiom)
mvalid@(mforall_prop@λa: \$i → \$o: (mimplies@(mbox@fool@a)@(mbox@fool@(mbox@fool@a))))   thf(k_axiom_for_fool, a
mvalid@(mforall_prop@λphi: \$i → \$o: (mimplies@(mbox@fool@phi)@(mbox@a@phi)))   thf(i_axiom_for_fool_a, axiom)
mvalid@(mbox@a@λx: \$i: (tpp@catalunya@spain))   thf(ax_1, axiom)
mvalid@(mbox@fool@λx: \$i: (ec@spain@france))   thf(ax_2, axiom)
mvalid@(mbox@a@λx: \$i: (ntpp@paris@france))   thf(ax_3, axiom)
mvalid@(mbox@a@λx: \$i: ∀z: reg: ((ntpp@france@z and ntpp@spain@z) ⇒ (pp@paris@z and pp@catalunya@z)))   thf(c

GEG012∧1.p Something about France, Spain, Paris, Catalunya
include('Axioms/LCL013ˆ0.ax')
include('Axioms/LCL014ˆ0.ax')
catalunya: reg   thf(catalunya, type)
france: reg   thf(france, type)
spain: reg   thf(spain, type)
paris: reg   thf(paris, type)

a: \$i → \$i → \$o   thf(a, type)
fool: \$i → \$i → \$o   thf(fool, type)
mvalid@(mforall_prop@λa: \$i → \$o: (mimplies@(mbox@fool@a)@a))   thf(t_axiom_for_fool, axiom)
mvalid@(mforall_prop@λa: \$i → \$o: (mimplies@(mbox@fool@a)@(mbox@fool@(mbox@fool@a))))   thf(k_axiom_for_fool, a
mvalid@(mforall_prop@λphi: \$i → \$o: (mimplies@(mbox@fool@phi)@(mbox@a@phi)))   thf(i_axiom_for_fool_a, axiom)
mvalid@(mbox@a@λx: \$i: (tpp@catalunya@spain))   thf(ax$_1$, axiom)
mvalid@(mbox@fool@λx: \$i: (ec@spain@france))   thf(ax$_2$, axiom)
mvalid@(mbox@a@λx: \$i: (ntpp@paris@france))   thf(ax$_3$, axiom)
mvalid@(mbox@a@λx: \$i: ∀x: reg: ((ntpp@france@x and ntpp@spain@x) ⇒ (ntpp@paris@x and ntpp@catalunya@x)))

GEG013∧1.p Unequal regions
include('Axioms/LCL013^0.ax')
include('Axioms/LCL014^0.ax')
catalunya: reg   thf(catalunya, type)
france: reg   thf(france, type)
spain: reg   thf(spain, type)
paris: reg   thf(paris, type)
a: \$i → \$i → \$o   thf(a, type)
fool: \$i → \$i → \$o   thf(fool, type)
mvalid@(mforall_prop@λa: \$i → \$o: (mimplies@(mbox@fool@a)@a))   thf(t_axiom_for_fool, axiom)
mvalid@(mforall_prop@λa: \$i → \$o: (mimplies@(mbox@fool@a)@(mbox@fool@(mbox@fool@a))))   thf(k_axiom_for_fool, a
mvalid@(mforall_prop@λphi: \$i → \$o: (mimplies@(mbox@fool@phi)@(mbox@a@phi)))   thf(i_axiom_for_fool_a, axiom)
mvalid@(mbox@a@λx: \$i: (tpp@catalunya@spain))   thf(ax$_1$, axiom)
mvalid@(mbox@fool@λx: \$i: (ec@spain@france))   thf(ax$_2$, axiom)
mvalid@(mbox@a@λx: \$i: (ntpp@paris@france))   thf(ax$_3$, axiom)
mvalid@(mbox@a@λx: \$i: ∃z: reg, y: reg: ¬ eq@z@y)   thf(con, conjecture)

GEG014∧1.p Two unequal regions in France
include('Axioms/LCL013^0.ax')
include('Axioms/LCL014^0.ax')
catalunya: reg   thf(catalunya, type)
france: reg   thf(france, type)
spain: reg   thf(spain, type)
paris: reg   thf(paris, type)
a: \$i → \$i → \$o   thf(a, type)
fool: \$i → \$i → \$o   thf(fool, type)
mvalid@(mforall_prop@λa: \$i → \$o: (mimplies@(mbox@fool@a)@a))   thf(t_axiom_for_fool, axiom)
mvalid@(mforall_prop@λa: \$i → \$o: (mimplies@(mbox@fool@a)@(mbox@fool@(mbox@fool@a))))   thf(k_axiom_for_fool, a
mvalid@(mforall_prop@λphi: \$i → \$o: (mimplies@(mbox@fool@phi)@(mbox@a@phi)))   thf(i_axiom_for_fool_a, axiom)
mvalid@(mbox@a@λx: \$i: (tpp@catalunya@spain))   thf(ax$_1$, axiom)
mvalid@(mbox@fool@λx: \$i: (ec@spain@france))   thf(ax$_2$, axiom)
mvalid@(mbox@a@λx: \$i: (ntpp@paris@france))   thf(ax$_3$, axiom)
mvalid@(mbox@a@λx: \$i: ∃z: reg, y: reg: (¬ eq@z@y and o@z@france and o@z@france))   thf(con, conjecture)

GEG015∧1.p Two unequal regions in France
include('Axioms/LCL013^0.ax')
include('Axioms/LCL014^0.ax')
catalunya: reg   thf(catalunya, type)
france: reg   thf(france, type)
spain: reg   thf(spain, type)
paris: reg   thf(paris, type)
a: \$i → \$i → \$o   thf(a, type)
fool: \$i → \$i → \$o   thf(fool, type)
mvalid@(mforall_prop@λa: \$i → \$o: (mimplies@(mbox@fool@a)@a))   thf(t_axiom_for_fool, axiom)
mvalid@(mforall_prop@λa: \$i → \$o: (mimplies@(mbox@fool@a)@(mbox@fool@(mbox@fool@a))))   thf(k_axiom_for_fool, a
mvalid@(mforall_prop@λphi: \$i → \$o: (mimplies@(mbox@fool@phi)@(mbox@a@phi)))   thf(i_axiom_for_fool_a, axiom)
mvalid@(mbox@a@λx: \$i: (tpp@catalunya@spain))   thf(ax$_1$, axiom)
mvalid@(mbox@fool@λx: \$i: (ec@spain@france))   thf(ax$_2$, axiom)
mvalid@(mbox@a@λx: \$i: (ntpp@paris@france))   thf(ax$_3$, axiom)
mvalid@(mbox@a@λx: \$i: ∃z: reg, y: reg: (¬ eq@z@y and p@z@france and p@y@france))   thf(con, conjecture)

GEG016∧1.p Places in Spain and France do not overlap
include('Axioms/LCL013^0.ax')
include('Axioms/LCL014^0.ax')
catalunya: reg thf(catalunya, type)
france: reg thf(france, type)
spain: reg thf(spain, type)
paris: reg thf(paris, type)
a: \$i → \$i → \$o thf(*a*, type)
fool: \$i → \$i → \$o thf(fool, type)
mvalid@(mforall_prop@λ*a*: \$i → \$o: (mimplies@(mbox@fool@*a*)@*a*)) thf(t_axiom_for_fool, axiom)
mvalid@(mforall_prop@λ*a*: \$i → \$o: (mimplies@(mbox@fool@*a*)@(mbox@fool@(mbox@fool@*a*)))) thf(k_axiom_for_fool, a
mvalid@(mforall_prop@λphi: \$i → \$o: (mimplies@(mbox@fool@phi)@(mbox@*a*@phi))) thf(i_axiom_for_fool_a, axiom)
mvalid@(mbox@*a*@λ*x*: \$i: (tpp@catalunya@spain)) thf(ax$_1$, axiom)
mvalid@(mbox@fool@λ*x*: \$i: (ec@spain@france)) thf(ax$_2$, axiom)
mvalid@(mbox@*a*@λ*x*: \$i: (ntpp@paris@france)) thf(ax$_3$, axiom)
mvalid@(mbox@fool@λ*x*: \$i: ∀*z*: reg, *y*: reg: ((*p*@*z*@spain and *p*@*y*@france) ⇒ ¬ *o*@*z*@*y*)) thf(con, conjecture)

GEG018∧1.p Some places are non-tangential proper parts
include('Axioms/LCL013^0.ax')
include('Axioms/LCL014^0.ax')
catalunya: reg thf(catalunya, type)
france: reg thf(france, type)
spain: reg thf(spain, type)
paris: reg thf(paris, type)
a: \$i → \$i → \$o thf(*a*, type)
fool: \$i → \$i → \$o thf(fool, type)
mvalid@(mforall_prop@λ*a*: \$i → \$o: (mimplies@(mbox@fool@*a*)@*a*)) thf(t_axiom_for_fool, axiom)
mvalid@(mforall_prop@λ*a*: \$i → \$o: (mimplies@(mbox@fool@*a*)@(mbox@fool@(mbox@fool@*a*)))) thf(k_axiom_for_fool, a
mvalid@(mforall_prop@λphi: \$i → \$o: (mimplies@(mbox@fool@phi)@(mbox@*a*@phi))) thf(i_axiom_for_fool_a, axiom)
mvalid@(mbox@*a*@λ*x*: \$i: (tpp@catalunya@spain)) thf(ax$_1$, axiom)
mvalid@(mbox@fool@λ*x*: \$i: (ec@spain@france)) thf(ax$_2$, axiom)
mvalid@(mbox@*a*@λ*x*: \$i: (ntpp@paris@france)) thf(ax$_3$, axiom)
mvalid@(mbox@fool@λ*x*: \$i: ¬ ∃*z*: reg: ∀*y*: reg: (ntpp@*z*@*y*)) thf(con, conjecture)

GEG019∧1.p If Paris and Catalunya overlap, then so do Spain and France
include('Axioms/LCL013^0.ax')
include('Axioms/LCL014^0.ax')
catalunya: reg thf(catalunya, type)
france: reg thf(france, type)
spain: reg thf(spain, type)
paris: reg thf(paris, type)
a: \$i → \$i → \$o thf(*a*, type)
fool: \$i → \$i → \$o thf(fool, type)
mvalid@(mforall_prop@λ*a*: \$i → \$o: (mimplies@(mbox@fool@*a*)@*a*)) thf(t_axiom_for_fool, axiom)
mvalid@(mforall_prop@λ*a*: \$i → \$o: (mimplies@(mbox@fool@*a*)@(mbox@fool@(mbox@fool@*a*)))) thf(k_axiom_for_fool, a
mvalid@(mforall_prop@λphi: \$i → \$o: (mimplies@(mbox@fool@phi)@(mbox@*a*@phi))) thf(i_axiom_for_fool_a, axiom)
mvalid@(mbox@*a*@λ*x*: \$i: (tpp@catalunya@spain)) thf(ax$_1$, axiom)
mvalid@(mbox@fool@λ*x*: \$i: (ec@spain@france)) thf(ax$_2$, axiom)
mvalid@(mbox@*a*@λ*x*: \$i: (ntpp@paris@france)) thf(ax$_3$, axiom)
mvalid@(mbox@*a*@λ*x*: \$i: ∀*z*: reg: ((*o*@*z*@paris and *o*@*z*@catalunya) ⇒ (*o*@*z*@spain and *o*@*z*@france))) thf(con, conject

GEG020∧1.p A place that overlaps with Paris or Catalunya
include('Axioms/LCL013^0.ax')
include('Axioms/LCL014^0.ax')
catalunya: reg thf(catalunya, type)
france: reg thf(france, type)
spain: reg thf(spain, type)
paris: reg thf(paris, type)
a: \$i → \$i → \$o thf(*a*, type)
fool: \$i → \$i → \$o thf(fool, type)
mvalid@(mforall_prop@λ*a*: \$i → \$o: (mimplies@(mbox@fool@*a*)@*a*)) thf(t_axiom_for_fool, axiom)

mvalid@(mforall_prop@λ*a*: $i → $o: (mimplies@(mbox@fool@*a*)@(mbox@fool@(mbox@fool@*a*)))) thf(k_axiom_for_fool, a
mvalid@(mforall_prop@λphi: $i → $o: (mimplies@(mbox@fool@phi)@(mbox@*a*@phi))) thf(i_axiom_for_fool_a, axiom)
mvalid@(mbox@*a*@λ*x*: $i: (tpp@catalunya@spain)) thf(ax_1, axiom)
mvalid@(mbox@fool@λ*x*: $i: (ec@spain@france)) thf(ax_2, axiom)
mvalid@(mbox@*a*@λ*x*: $i: (ntpp@paris@france)) thf(ax_3, axiom)
mvalid@(mbox@*a*@λ*x*: $i: ∀*z*: reg: ((*o*@*z*@paris or *o*@*z*@catalunya) ⇒ (*o*@*z*@spain or *o*@*z*@france))) thf(con, conjecture

GEG021=1.p Estimate distance between cities (one step)
city: $tType tff(city_type, type)
d: (city × city) → $int tff(d_type, type)
kiel: city tff(kiel_type, type)
hamburg: city tff(hamburg_type, type)
berlin: city tff(berlin_type, type)
cologne: city tff(cologne_type, type)
frankfurt: city tff(frankfurt_type, type)
saarbruecken: city tff(saarbruecken_type, type)
munich: city tff(munich_type, type)
(∀*x*: city, *y*: city: *d*(*x*, *y*) = *d*(*y*, *x*) and ∀*x*: city, *y*: city, *z*: city: $lesseq(*d*(*x*, *z*), $sum(*d*(*x*, *y*), *d*(*y*, *z*))) and ∀*x*: city: *d*(*x*, *x*) = 0 and *d*(berlin, munich) = 510 and *d*(berlin, cologne) = 480 and *d*(berlin, frankfurt) = 420 and *d*(saarbruecken, frankfurt) = 160 and *d*(saarbruecken, cologne) = 190 and *d*(hamburg, cologne) = 360 and *d*(hamburg, frankfurt) = 390 and *d*(cologne, fran
150 and *d*(hamburg, kiel) = 90 and *d*(hamburg, berlin) = 250 and *d*(munich, frankfurt) = 300 and *d*(munich, saarbruecken) =
360) ⇒ $lesseq(*d*(cologne, berlin), 500) tff($\mathrm{city_distance}_1$, conjecture)

GEG022=1.p Estimate distance between cities (two steps)
city: $tType tff(city_type, type)
d: (city × city) → $int tff(d_type, type)
kiel: city tff(kiel_type, type)
hamburg: city tff(hamburg_type, type)
berlin: city tff(berlin_type, type)
cologne: city tff(cologne_type, type)
frankfurt: city tff(frankfurt_type, type)
saarbruecken: city tff(saarbruecken_type, type)
munich: city tff(munich_type, type)
(∀*x*: city, *y*: city: *d*(*x*, *y*) = *d*(*y*, *x*) and ∀*x*: city, *y*: city, *z*: city: $lesseq(*d*(*x*, *z*), $sum(*d*(*x*, *y*), *d*(*y*, *z*))) and ∀*x*: city: *d*(*x*, *x*) = 0 and *d*(berlin, munich) = 510 and *d*(berlin, cologne) = 480 and *d*(berlin, frankfurt) = 420 and *d*(saarbruecken, frankfurt) = 160 and *d*(saarbruecken, cologne) = 190 and *d*(hamburg, cologne) = 360 and *d*(hamburg, frankfurt) = 390 and *d*(cologne, fran
150 and *d*(hamburg, kiel) = 90 and *d*(hamburg, berlin) = 250 and *d*(munich, frankfurt) = 300 and *d*(munich, saarbruecken) =
360) ⇒ $lesseq(*d*(hamburg, munich), 700) tff($\mathrm{city_distance}_2$, conjecture)

GEG023=1.p Estimate distance between cities (three steps)
city: $tType tff(city_type, type)
d: (city × city) → $int tff(d_type, type)
kiel: city tff(kiel_type, type)
hamburg: city tff(hamburg_type, type)
berlin: city tff(berlin_type, type)
cologne: city tff(cologne_type, type)
frankfurt: city tff(frankfurt_type, type)
saarbruecken: city tff(saarbruecken_type, type)
munich: city tff(munich_type, type)
(∀*x*: city, *y*: city: *d*(*x*, *y*) = *d*(*y*, *x*) and ∀*x*: city, *y*: city, *z*: city: $lesseq(*d*(*x*, *z*), $sum(*d*(*x*, *y*), *d*(*y*, *z*))) and ∀*x*: city: *d*(*x*, *x*) = 0 and *d*(berlin, munich) = 510 and *d*(berlin, cologne) = 480 and *d*(berlin, frankfurt) = 420 and *d*(saarbruecken, frankfurt) = 160 and *d*(saarbruecken, cologne) = 190 and *d*(hamburg, cologne) = 360 and *d*(hamburg, frankfurt) = 390 and *d*(cologne, fran
150 and *d*(hamburg, kiel) = 90 and *d*(hamburg, berlin) = 250 and *d*(munich, frankfurt) = 300 and *d*(munich, saarbruecken) =
360) ⇒ $lesseq(*d*(kiel, saarbruecken), 640) tff($\mathrm{city_distance}_3$, conjecture)

GEG024=1.p Find sufficiently large and sufficiently close city (easy)
city: $tType tff(city_type, type)
d: (city × city) → $int tff(d_type, type)
inh: city → $int tff(inh_type, type)
kiel: city tff(kiel_type, type)
hamburg: city tff(hamburg_type, type)

berlin: city   tff(berlin_type, type)
cologne: city   tff(cologne_type, type)
frankfurt: city   tff(frankfurt_type, type)
saarbruecken: city   tff(saarbruecken_type, type)
munich: city   tff(munich_type, type)
($\forall x$: city, y: city: $d(x, y) = d(y, x)$ and $\forall x$: city, y: city, z: city: \$lesseq($d(x, z)$, \$sum($d(x, y), d(y, z)$)) and $\forall x$: city: $d(x, x) =$
0 and d(berlin, munich) = 510 and d(berlin, cologne) = 480 and d(berlin, frankfurt) = 420 and d(saarbruecken, frankfurt) =
160 and d(saarbruecken, cologne) = 190 and d(hamburg, cologne) = 360 and d(hamburg, frankfurt) = 390 and d(cologne, fran
150 and d(hamburg, kiel) = 90 and d(hamburg, berlin) = 250 and d(munich, frankfurt) = 300 and d(munich, saarbruecken) =
360 and inh(berlin) = 3442675 and inh(hamburg) = 1774224 and inh(munich) = 1330440 and inh(cologne) = 998105 and inh
671927 and inh(saarbruecken) = 175810 and inh(kiel) = 238281) ⇒ $\exists x$: city: (\$lesseq(d(kiel, x), 100) and \$lesseq(1000000, in

GEG025=1.p Find sufficiently large and sufficiently close city (medium)
city: \$tType   tff(city_type, type)
d: (city × city) → \$int   tff(d_type, type)
inh: city → \$int   tff(inh_type, type)
kiel: city   tff(kiel_type, type)
hamburg: city   tff(hamburg_type, type)
berlin: city   tff(berlin_type, type)
cologne: city   tff(cologne_type, type)
frankfurt: city   tff(frankfurt_type, type)
saarbruecken: city   tff(saarbruecken_type, type)
munich: city   tff(munich_type, type)
($\forall x$: city, y: city: $d(x, y) = d(y, x)$ and $\forall x$: city, y: city, z: city: \$lesseq($d(x, z)$, \$sum($d(x, y), d(y, z)$)) and $\forall x$: city: $d(x, x) =$
0 and d(berlin, munich) = 510 and d(berlin, cologne) = 480 and d(berlin, frankfurt) = 420 and d(saarbruecken, frankfurt) =
160 and d(saarbruecken, cologne) = 190 and d(hamburg, cologne) = 360 and d(hamburg, frankfurt) = 390 and d(cologne, fran
150 and d(hamburg, kiel) = 90 and d(hamburg, berlin) = 250 and d(munich, frankfurt) = 300 and d(munich, saarbruecken) =
360 and inh(berlin) = 3442675 and inh(hamburg) = 1774224 and inh(munich) = 1330440 and inh(cologne) = 998105 and inh
671927 and inh(saarbruecken) = 175810 and inh(kiel) = 238281) ⇒ $\exists x$: city: (\$lesseq(d(saarbruecken, x), 600) and \$lesseq(3