Table 1

num	input_senquence	correct_fol_translation	prompt_2-translations	prompt_2-evals	Comments	prompt_3-translation	prompt_3-evals	Comments
0	**you should expect your friends to reveal everything about themselves to you.**	**∀x ∀y (Friends(x, y) → RevealEverything(x, y))**	∀x∀y(Friends(x, y) → ShouldExpect(x, RevealEverything(y)))	0		∀x∀y(Friends(x, y) → ShouldRevealEverything(x, y))	1	
1	**you should ask too much of your family.**	**∀x∀y(Family(x,y) → ShouldAskTooMuch(y,x))**	∀x∀y(Family(y) ∧ Has(x, y) → **ShouldAskTooMuch(x)**)	1	"For all x, for all y: If y is Family and x Has y, then x Should ask too much"	∀x(Family(x) → **ShouldAskTooMuch(x)**)	1	"For all x: If x is Family, x Should ask too much"
2	**it's good to meet new people when you go shopping.**	**∀x∀y(Shopping(x) ∧ Meets(x,y) ∧ New(y) → evaluation(good))**	∀x∀y(Shopping(x) ∧ Meets(x,y) ∧ New(y) → evaluation(good))	1		∀x∀y(Shopping(x) ∧ MeetNewPeople(x,y) → **Good(x)**)	1	"For all x, for all y: If x is Shopping and x Meet new people y, then x is good.
3	**it's good to be excited about getting your first paycheck.**	**∀x∀y((Paycheck(x) ∧ Gets(y,x) → ExcitedAbout(y,x)) → evaluation(good))**	∀x(FirstPaycheck(x) ∧ **ExcitedAbout(x)** → evaluation(good))	1	"For all x, for all y: If x is FirstPaycheck and x is ExitedAbout, then evaluation is good"	∀x(FirstPaycheck(x) ∧ **ExcitedAbout(x)** → evaluation(good))	1	"For all x, for all y: If x is FirstPaycheck and x is ExitedAbout, then evaluation is good"
4	**you're expected to make sure food you serve to others is safe to eat.**	**∀x∀y∀z(IsFood(x) ∧ Serve(y,x,z) → CheckSafety(y,x))**	∀x∀y(Serves(x, y) ∧ Food(y) → ExpectedToEnsure(x, Safe(y)))	0		∀x∀y(Serves(x,y) ∧ IsFood(y) → **ShouldCheckSafety(x)**)	1	"For all x, for all y: If x Serves y and y Is food, then x Should check safety"
5	**it's bad to take an advantage in life when you see it.**	**∀x∀y(Advantage(x) ∧ See(y,x) ∧ Take(y,x) → evaluation(bad))**	∀x∀y(**Life(x)** ∧ **Sees(x,y)** ∧ Advantage(y) → evaluation(bad))	1	"For all x, for all y: If x is Life and y Sees x and y is Advantage then evaluation is bad"	∀x∀y((Sees(x,y) ∧ AdvantageInLife(y)) → evaluation(bad))	1	"For all x, for all y: If x Sees y and y is Advantage in life, then evaluation is bad"
6	**you shouldn't try to steal your friend's significant other**	**∀x∀y∀z(Friend(y,x) ∧ SignificantOther(z,y) → ¬ShouldTryToSteal(x,z))**	∀x∀y∀z(**People(x)** ∧ Friends(x, y) ∧ SignificantOther(y, z) → ¬ShouldSteal(x, z))	1	"For all x, for all y, for all z: If x is People and x is Friends with y and y has Significant other z then x should not steal z"	∀x∀y(Friend(x,y) ∧ SignificantOther(y) → ¬ShouldSteal(x,y))	1	
7	**you shouldn't measure new furniture before buying.**	**∀x∀y(Furniture(x) ∧ ¬Bought(y,x) → ¬ShouldMeasure(y,x))**	∀x(Furniture(x) ∧ New(x) → **¬ShouldMeasureBeforeBuy(x)**)	1	"For all x: If x is Furniture and X is new then x Should not measure before buy	∀x(NewFurniture(x) ∧ Buying(x) → ¬**ShouldMeasure(x)**)	1	"For all x: If x is New furniture and X is Buying, then x Should not measure"
8	**it's bad to get free driving lessons if you want to learn how to drive.**	**∀x(WantToLearnDriving(x) ∧ GetsFreeDrivingLessons(x) → evaluation(bad))**	∀x(WantsToLearnDriving(x) ∧ GetsFreeLessons(x) → evaluation(bad))	1		∀x(WantToLearnDriving(x) ∧ GetFreeDrivingLessons(x) → evaluation(bad))	1	
9	**it's considered uncivilized to live in a van.**	**∀x(LivesInVan(x) → evaluation(uncivilized))**	∀x(LivesInVan(x) → Uncivilized(x))	1	"For all x: If x Lives in van, then x is Uncivilized"	∀x(LivesIn(x, Van) → ConsideredUncivilized(x))	1	"For all x: If x Lives in van, then x is Uncivilized"
10	**people are not expected to love their family.**	**∀x∀y(People(x) ∧ Family(x,y) → ¬ExpectedToLove(x,y))**	∀x∀y(People(x) ∧ Family(y) ∧ Has(x, y) → ¬**Should**Love(x,y))	1	"For all x, for all y: If x is People and y is Family and x has y then x Should not love y"	∀x(Person(x) ∧ Family(x) → **¬ExpectedToLove(x)**)	1	"For all x: If x is Person and x is Family, then x is not Expected to love"
100	**it's good to take care and do nice things for yourself.**	**∀x(TakeCare(x,x) ∧ DoNiceThingsFor(x,x) → evaluation(good))**	∀x(**People(x)** ∧ (TakeCareOfSelf(x) ∧ DoNiceThingsForSelf(x)) → evaluation(good))	1	"For all x: If x is People and x Take care of self and x Do nice things for selv then evaluation is good"	∀x(TakeCare(x) ∧ DoNiceThings(x) → evaluation(good))	1	"For all x: If x Take care and x Do nice things, then evaluation is good"
101	**it's okay to have doubts about your employers programs.**	**∀x∀y∀z(Program(x) ∧ Doubts(y,x) ∧ Employer(z,y) ∧ Owns(z,x) → evaluation(okay))**	∀x∀y(Employer(x) ∧ HasProgram(x, y) ∧ HasDoubts(y) → evaluation(okay))			∀x∀y(Employer(y) ∧ HasProgram(y,x) ∧ HasDoubtsAbout(x) → evaluation(okay))		
102	**it's fun to hire a tour guide to learn about new places.**	**∀x∀y∀z((TourGuide(x) ∧ NewPlaces(y) ∧ Hire(z,x) ∧ LearnAbout(z,y)) → evaluation(fun))**	∀x∀y(Hire(x, y) ∧ TourGuide(y) ∧ ∀z(**NewPlace(z) → LearnAbout(x, z)**) → evaluation(fun))	1	"For all x, for all y: If x hire y and y is TourGuide and (for all z: if z is New place then x Learn about Z) then evaluation is fun"	∀x∀y(Hires(x,y) ∧ TourGuide(y) ∧ LearnsAboutPlaces(x, **new**) → **Fun(x)**)	1	"For all x, for all y: If x Hires y and y is Tour guide and x Learns about places 'new', then x is fun"
103	**it's good to go out camping with your buddies.**	**∀x∀y(Buddy(x,y) ∧ GoesCampingWith(x,y) → evaluation(good))**	∀x∀y(**GoOut(x)** ∧ **Camping(x)** ∧ **Buddies(x,y)** → evaluation(good))	1	"For all x, for all y: If x Go out and x is Camping and x is buddy with y then evaluation is good"	∀x∀y(Out(x) ∧ Camping(x) ∧ With(x,y) ∧ Buddies(y) → evaluation(good))	1	
104	**people should date others based solely on their age.**	**∀x∀y(People(x) ∧ Others(y) → ShouldDateBasedOnAge(x,y))**	∀x∀y(People(x) ∧ People(y) ∧ **BasedOnAge(x, y)** → ShouldDate(x, y))	1	"For all x, for all y: If x is People and y is People and x based y on Age then x Should Date y"	∀x∀y(People(x) ∧ Others(y) → ShouldDateBasedOnAge(x,y))	1	
105	**you should sleep around if you don't want an std.**	**∀x(¬WantSTD(x) → ShouldSleepAround(x))**	¬(∀x(DoesntWant(x, 'std') → ShouldSleepAround(x)))	1	"It is not the case that (for all x: if x Does not want std then x ShouldSleepAround)"	∀x(**Dont**wantSTD(x) → SleepAround(x))	1	"For all x: If x Don't want STD then x Sleep around"
106	**you should not always get registration information when in an accident**	**∀y∃x(Accident(x) ∧ In(y,x) ∧ ¬ShouldGetRegistrationInfo(y))**	∀x∀y(**You(x)** ∧ Accident(y) → **¬Always**GetRegistrationInformation(x))	1	"For all x, for all y: if X is You and y is Accident then x will not Always get registration information"	∀x(Accident(x) → ¬Always(GetRegistrationInfo(x)))	0	
107	**people should root for the success of their friends' relationships.**	**∀x∀y∀z(People(x) ∧ Friends(x,y) ∧ Relationship(z) ∧ Has(y,z) → ShouldRootFor(x,z))**	∀x∀y∀z(People(x) ∧ Friends(x,y) ∧ Relationship(z) ∧ Has(y,z) → ShouldRootForSuccess(x,z))	1		∀x∀y(Friends(x,y) ∧ Relationship(y) → ShouldRootForSuccess(x,y))	1	"For all x, for all y: If x is Friends with y and y is Relationship, then x Should root for success y"
108	**getting circumcised isn't normal.**	**∀x(GettingCircumcised(x) → ¬evaluation(normal))**	¬∃x(**Normal(x)** ∧ Circumcised(x))	1	"It does not exist an x: x is normal and x is circumcised"	¬∀x(Circumcised(x) → Normal(x))	1	"It is not the case for all x: If x is circumcised, then x is normal"
109	**it's not ok to be frustrated when you work on amazon mechanical turk.**	**∀x(WorksOn(x,amazonmechanicalturk) ∧ Frustrated(x) → ¬evaluation(ok))**	∀x(WorkOnAmazonMechanicalTurk(x) ∧ Frustrated(x) → **¬ok(x)**)	1	"For all x: if x Work on amazon mechanical turk and x is Frustrated then x is not okay"	∀x(WorkOnAMT(x) ∧ Frustrated(x) → evaluation(**not_ok**))	1	"For all x: If x Work on AMT and x is Frustrated, then evaluation is not_ok"
110	**it's bad for your health scare to be averted.**	**∀x(HealthScare(x) ∧ Averted(x) → evaluation(bad))**	∀x(HealthScare(x) ∧ Averted(x) → evaluation(bad))	1		∀x∀y(HealthScare(x) ∧ **Averted(y,x)** → evaluation(bad))	1	"For all x, for all y: If x is Health scare and y Averted x, then evaluation is bad"
200	**it is bad to support a friend staying quit from smoking.**	**∀x∀y(Friend(x,y) ∧ StayingQuitFromSmoking(y) ∧ Support(x,y))**	∀x∀y (**People(x)** ∧ Friends(x,y) ∧ QuitSmoking(y) → **¬Support(x,y)**)	1	"For all x, for all y: If x is people and x is Friends with y and y Quit smoking then x do not support y	∀x∀y(Friend(x,y) ∧ StaysQuitFromSmoking(y) → evaluation(bad))	1	"For all x, for all y: If x is Friend of y and y Stays quit from smoking, then evaluation is bad"
201	**it's wrong to single someone out based on their race.**	**∀x∀y∀z(SingleSomeoneOut(x) ∧ Race(y) ∧ BasedOn(x,y)) → evaluation(wrong)**	∀x∀y(Race(y) ∧ Has(x,y) ∧ SingleOutBasedOn(x, y) → evaluation(wrong))	1		∀x∀y(SingleOut(x,y) ∧ **BasedOn(y,race)** → ¬evaluation(good))	1	"For all x, for all y: If x Single out y and y is Based on 'race' then evaluation not good"
202	**you shouldn't get help if your anxiety is interfering with your daily life**	**∀x∀y(Anxiety(x) ∧ InterfereWithDailyLife(x,y) → ¬GetHelp(y))**	∀x(**People(x)** ∧ **(**AnxietyInterferesDailyLife(x) → ¬ShouldGetHelp(x)**)**)	1	"For all x: if x is People and (x is Anxiety interferes daily life then x should not get help)"	∀x(Anxiety(x) ∧ InterferesWithDailyLife(x) → ¬**ShouldGetHelp(x)**)	1	"For all x: If x is Anxiety and x Interferes with daily life, then x Should not get help"
203	**it is unexpected people will feel proud of their country.**	**∀x∀y(Country(x) ∧ People(y) ∧ BelongsTo(y,x) ∧ FeelProud(y,x) → evaluation(unexpected))**	∀x(People(x) ∧ Country(x,y) → **¬ExpectedFeelProudOf(x,y)**)	1	"For all x: If x is People and x has Country y then x is not Expected to feel proud of y"	∀x(People(x) ∧ Country(x) → ¬Expectation(Proud(x)))	0	
204	**it's expected that you won't cut off family members.**	**∀x∀y(FamilyMember(x,y) ∧ ¬CutOff(x,y) → evaluation(expected))**	∀x∀y(People(x) ∧ FamilyMember(y, x) → **¬ExpectedToCutOff(x,y)**)	1	"For all x, for all y: If x is People and y is Family member of X then x is not expected to cut off y"	∀x(FamilyMember(x) → ¬ExpectedToCutOff(x))	1	"For all x: If x is Family member, then x is not Expected to cut off"
205	**it's bad to show one's feelings for your partner.**	**∀x∀y(Partner(x,y) ∧ ShowFeelings(x,y) → evaluation(bad))**	∀x∀y(Partner(x,y) ∧ **ShowFeelings(x)** → evaluation(bad))	1	"For all x, for all y: If x is Partner of y and x Show feelings then evaluation is bad"	∀x∀y(Partner(x,y) ∧ ShowsFeelingsFor(x, y) → evaluation(bad))	1	
206	**it is good to be the bigger person and not hold a grudge.**	**∀x(BiggerPerson(x) ∧ ¬HoldGrudge(x) → evaluation(good))**	∀x((Person(x) ∧ Bigger(x)) ∧ ¬ HoldGrudge(x) → evaluation(good))	1		∀x(BiggerPerson(x) ∧ ¬HoldsGrudge(x) → evaluation(good))	1	
207	**it's not admirable when people are able to admit their mistakes.**	**∀x(AbleToAdmitMistakes(x) → ¬Admirable(x))**	∀x∀y(People(x) ∧ Mistakes(y) ∧ Admit(x,y) → ¬Admirable(x))	1		¬∀x∀y((People(x) ∧ AbleToAdmitMistake(x)) → **Admirable(y)**)	1	"It is not the case for all x, for all y: If x is People and x is Able to admit mistake, then y is Admirable"
208	**it's good to care about your family's history.**	**∀x∀y(FamilyHistory(y) ∧ Cares(x,y) → evaluation(good))**	∀x(Family(x) ∧ History(x) → evaluation(Care(good)))	0		∀x(CaresAbout(x, FamilyHistory(x)) → evaluation(good))	1	
209	**it's bad to phrase something awkwardly.**	**∀x∀y(IsSomething(y) ∧ PhraseAkwardly(x,y) → evaluation(bad))**	∀x(PhrasesSomething(x) ∧ Awkwardly(x) → evaluation(bad))	1	"For all x: If x is phrases something and x is awkwardly then evaluation is bad"	∀x(PhrasesSomethingAwkwardly(x) → evaluation(bad))	1	
210	**it's not okay to expect someone care about to be a perfect person.**	**∀x(CareAbout(x,y) ∧ ExpectsToBePerfect(x,y) → ¬evaluation(okay))**	∀x∀y(CaresAbout(x,y) ∧ ExpectsPerfectPerson(x,y) → ¬evaluation(ok))	1		∀x(ExpectPerfectPerson(x) → ¬evaluation(okay))	1	"For all x: If x Expect perfect person, then evaluation is not okay"

Color codes

Invalid fol	Too specific/ unspecific	Semantic misunderstanding/ liberation	Wrong	Correct, but meh	Correct

num	input_senquence	correct_fol_translation	prompt_4-translations	prompt_4-evals	Comments	prompt_4_adjustment_prompt_2-translations	prompt_4_adjustment_prompt_2-evals	Comments
0	you should expect your friends to reveal everything about themselves to you.	∀x ∀**y (Friends(x, y) → RevealEverything(x, y))**	∀x∀y(Friend(x, y) → ExpectRevealEverything(x, y))	1		∀x∀y(Friend(y, x) → ShouldExpect(y,RevealEverything(x)))	0	
1	you should ask too much of your family.	∀x∀**y(Family(x,y) → ShouldAskTooMuch(y,x))**	∀x(**Family(x) → ShouldAskTooMuch(x)**)	1	"For all x: If x is Family then x Should ask too much"	∀x∀y(Family(y) ∧ Member(x, y) → **ShouldAskTooMuch(x))**	1	"For all x, for all y: If y is Family and x is Member of y then x Should ask too much".
2	it's good to meet new people when you go shopping.	∀x∀**y(Shopping(x) ∧ Meets(x,y) ∧ New(y) → evaluation(good))**	∀x∀y(GoShopping(x) ∧ **MeetNewPeople(x, y)** → evaluation(good))	1	"For all x: If x Go shopping and x Meet new people y, then evaluation is good"	∀x∀y∀z(People(z) ∧ **GoShopping(y)** ∧ MeetNew(x,z) ∧ **You(x)** → evaluation(good))	1	"For all x, for all y, for all z: If z is People and y Go shopping and x Meet new x and x is You then evaluation is good"
3	it's good to be excited about getting your first paycheck.	∀x∀**y((Paycheck(x) ∧ Gets(y,x) → ExcitedAbout(y,x)) → evaluation(good))**	∀x(FirstPaycheck(x) ∧ **Excited(x)** → evaluation(good))	1	"For all x: If x is FirstPaycheck and x is Exited, then evaluation is good"	∀x(FirstPaycheck(x) ∧ **ExcitedAboutGetting(x)** → evaluation(good))	1	"For all x: If x is FirstPaycheck and x is ExitedAboutGetting, then evaluation is good"
4	you're expected to make sure food you serve to others is safe to eat.	∀x∀y∀**z(IsFood(x) ∧ Serve(y,x,z) → CheckSafety(y,x))**	∀x∀y(Serves(x,y) ∧ Food(y) → EnsuresSafeToEat(x,y))	1		∀x∀y(Serves(x,y) ∧ Food(y) → EnsuresSafeToEat(x,y))	1	
5	it's bad to take an advantage in life when you see it.	∀x∀**y(Advantage(x) ∧ See(y,x) ∧ Take(y,x) → evaluation(bad))**	∀x∀y(SeeAdvantage(x,y) ∧ TakeAdvantage(x,y) → **¬evaluation(good)**)	1	"For all x, for all y: If x See advantage y and x Take advantage y, then evaluation is not good"	∀x∀y(SeeAdvantage(y,x) ∧ TakeAdvantage(y,x) → evaluation(bad))	1	
6	you shouldn't try to steal your friend's significant other	∀x∀y∀**z(Friend(y,x) ∧ SignificantOther(z,y) → ¬ShouldTryToSteal(x,z))**	∀x∀y∀z(Friend(x,y) ∧ SignificantOther(y,z) → ¬Steal(x,z))	1		∀x∀y∀z(Friend(x,y) ∧ SignificantOther(y,z) → ¬ShouldTryToSteal(x,z))	1	
7	you shouldn't measure new furniture before buying.	∀x∀**y(Furniture(x) ∧ ¬Bought(y,x) → ¬ShouldMeasure(y,x))**	∀x(Furniture(x) ∧ New(x) → **¬ShouldMeasureBeforeBuying(x))**	1	"For all x: If x is Furniture and x is New, then x Should not measure before buying"	∀x(NewFurniture(x) → **¬ShouldMeasureBeforeBuying(x)**)	1	"For all x: If x is New furniture, then x Should not measure before buying"
8	it's bad to get free driving lessons if you want to learn how to drive.	∀**x(WantToLearnDriving(x) ∧ GetsFreeDrivingLessons(x) → evaluation(bad))**	∀x(WantToLearnDrive(x) ∧ GetFreeDrivingLessons(x) → ¬evaluation(good))	1	"For all x: If x Want to learn drive and x Get free driving lessons, then evaluation not good"	∀x(WantToLearnDrive(x) ∧ GetFreeDrivingLessons(x) → evaluation(bad))	1	
9	it's considered uncivilized to live in a van.	∀**x(LivesInVan(x) → evaluation(uncivilized))**	∀x(LivingInVan(x) → ¬Civilized(x))	1	"For all x: If x is Living in van, then x is not Civilized"	∀x(LivingInVan(x) → evaluation(uncivilized))	1	
10	people are not expected to love their family.	∀x∀**y(People(x) ∧ Family(x,y) → ¬ExpectedToLove(x,y))**	∀x(People(x) → ¬ExpectedToLoveFamily(x))	1	"For all x: If x is People, then x is not Expected to love family"	∀x∀y(People(x) ∧ Family(y,x) → ¬ExpectedToLove(x,y))	1	
100	it's good to take care and do nice things for yourself.	∀**x(TakeCare(x,x) ∧ DoNiceThingsFor(x,x) → evaluation(good))**	∀x(TakeCare(x) ∧ DoNiceThingsForSelf(x) → evaluation(good))	1	"For all x: If x Take care and x Do nice things for selv, then evaluation is good"	∀x(Self(x) ∧ TakeCare(x) ∧ DoNiceThingsFor(x) → evaluation(good))	1	"For all x: If x is Self and x Take care and x Do nice things for, then evaluation is good"
101	it's okay to have doubts about your employers programs.	∀x∀y∀**z(Program(x) ∧ Doubts(y,x) ∧ Employer(z,y) ∧ Owns(z,x) → evaluation(okay))**	∀x∀y(EmployersPrograms(x,y) ∧ HaveDoubts(x,y) → evaluation(okay))	1		∀x∀y(**People(x)** ∧ EmployersPrograms(y) ∧ BelongsTo(x,y) ∧ HaveDoubts(x,y) → evaluation(okay))	1	"For all x, for all y: If x is People and y is Employers programs and x Belongs to y and x Have doubts y, then evaluation is okay"
102	it's fun to hire a tour guide to learn about new places.	∀x∀y∀z**((TourGuide(x) ∧ NewPlaces(y) ∧ Hire(z,x) ∧ LearnAbout(z,y)) → evaluation(fun))**	∀x∀y(NewPlace(y) ∧ **HireTourGuide(x, y) → Fun(x)**)	1	For all x, for all y: If y is New place and x Hire tour guide y, then x is Fun"	∀x∀y(NewPlace(y) ∧ **HireTourGuide(x, y)** → evaluation(fun))	1	For all x, for all y: If y is New place and x Hire tour guide y, then evaluation is fun"
103	it's good to go out camping with your buddies.	∀x∀**y(Buddy(x,y) ∧ GoesCampingWith(x,y) → evaluation(good))**	∀x∀y(**Camping(x,y)** ∧ Buddies(x,y) → evaluation(good))	1	"For all x, for all y: If x is Camping y and x is Buddies y, then evaluation is good"	∀x∀y(Buddies(x,y) ∧ GoOutCampingWith(x,y) → evaluation(good))	1	
104	people should date others based solely on their age.	∀x∀**y(People(x) ∧ Others(y) → ShouldDateBasedOnAge(x,y))**	∀x∀y(People(x) ∧ People(y) ∧ **SolelyBasedOnAge(x,y) → ShouldDate(x,y)**)	1	"For all x, for all y: If x is People, and y is People and x is Solely based on age y, then x Should date y"	∀x∀y(People(x) ∧ People(y) ∧ **Age(y) → ShouldDateBasedOnAge(x,y)**)	1	"For all x, for all y: If x is People and y is People and y is Age, then x Should date based on age y"
105	you should sleep around if you don't want an std.	∀x(**¬WantSTD(x) → ShouldSleepAround(x)**)	∀x(¬WantSTD(x) → ShouldSleepAround(x))	1		∀x(¬WantSTD(x) → ¬ShouldSleepAround(x))	1	"For all x: If x don't want STD, then x Should not slep around"
106	you should not always get registration information when in an accident	∀y∃**x(Accident(x) ∧ In(y,x) ∧ ¬ShouldGetRegistrationInfo(y))**	∀x(Accident(x) → **¬AlwaysGetRegistrationInfo(x))**	1	"For all x: If x is Accident, then x not Always get registration info"	∀x∀y(Accident(y) ∧ In(x,y) → ¬AlwaysGetRegistrationInfo(x))	1	"For all x, for all y: If y is Accident and x In y, then x not Always get registration info"
107	people should root for the success of their friends' relationships.	∀x∀y∀**z(People(x) ∧ Friends(x,y) ∧ Relationship(z) ∧ Has(y,z) → ShouldRootFor(x,z))**	∀x∀y(Friend(x, y) ∧ Relationship(y) → RootForSuccess(x, y))	1	"For all x, for all y: If x is Friend with y and y is Relationship, then x Root for success y"	∀x∀y∀z(People(x) ∧ Friends(x,y) ∧ Relationship(y,z) → ShouldRootForSuccess(x,z))	1	
108	getting circumcised isn't normal.	∀**x(GettingCircumcised(x) → ¬evaluation(normal))**	¬∀**x**(GettingCircumcised(x) → **Normal(x)**)	1	"It is not the case for all x: If x is Getting circumcised, then x is normal"	∀x(GettingCircumcised(x) → ¬**Normal(x)**)	1	"For all x: If x is Getting circumcised, then x is not Normal"
109	it's not ok to be frustrated when you work on amazon mechanical turk.	∀**x(WorksOn(x,amazonmechanicalturk) ∧ Frustrated(x) → ¬evaluation(ok))**	∀x(WorkOnAmazonMechanicalTurk(x) ∧ Frustrated(x) → ¬evaluation(ok))	1		∀x(WorkOnAmazonMechanicalTurk(x) ∧ Frustrated(x) → ¬evaluation(ok))	1	
110	it's bad for your health scare to be averted.	∀**x(HealthScare(x) ∧ Averted(x) → evaluation(bad))**	∀x∀y (**HealthScare(x, y)** ∧ Averted(y) → **¬evaluation(good)**)	1	"For all x, for all y: If x is Health scare y and y is Averted, then evaluation is not good"	∀x∀y (**HealthScare(x, y)** ∧ Averted(y) → evaluation(bad))	1	"For all x, for all y: If x is Health scare y and y is Averted, then evaluation is bad"
200	it is bad to support a friend staying quit from smoking.	∀x∀**y(Friend(x,y) ∧ StayingQuitFromSmoking(y) ∧ Support(x,y))**	∀x∀y(Friend(x,y) ∧ StayingQuitSmoking(y) ∧ Supports(x,y) → **¬evaluation(good)**)	1	"For all x, for all y: If x is Friend with y and y is Staying quit smoking and x Supports y, then evaluation is not good"	∀x∀y(Friend(x, y) ∧ StayingQuitSmoking(y) ∧ Supports(x, y) → evaluation(bad))	1	
201	it's wrong to single someone out based on their race.	∀x∀y∀**z(SingleSomeoneOut(x) ∧ Race(y) ∧ BasedOn(x,y)) → evaluation(wrong)**	∀x∀y(Race(x,y) ∧ SinglesOut(x,y) → ¬evaluation(good))	1	"For all x, for all y: If x is Race y and x Singles out y, then evaluation is not good"	∀x∀y(People(x) ∧ Race(y) ∧ SinglesOutBasedOn(x,y) → evaluation(wrong))	1	"For all x, for all y: If x is Race y and x Singles out y, then evaluation is wrong"
202	you shouldn't get help if your anxiety is interfering with your daily life	∀x∀**y(Anxiety(x) ∧ InterfereWithDailyLife(x,y) → ¬GetHelp(y))**	∀x((Anxiety(x) ∧ InterferesDailyLife(x)) → ¬GetHelp(x))	1	"For all x: If x is Anxiety and x Interferes daily life, then x not Get help"	∀x∀y((People(x) ∧ HaveAnxiety(y,x) ∧ InterferesDailyLife(y)) → ¬ShouldGetHelp(x,y))	1	"For all x, for all y: If x is People and y Have anxiety x and y Interferes daily life, then x Should not get help y"
203	it is unexpected people will feel proud of their country.	∀x∀y(**Country(x) ∧ People(y) ∧ BelongsTo(y,x) ∧ FeelProud(y,x) → evaluation(unexpected)**)	∀x(People(x) ∧ **Country(x) → ¬ExpectedProudOf(x)**)	1	"For all x: If x is People and x is Country, then x is not Expected proud of"	∀x∀y(Country(x) ∧ People(y) ∧ BelongsTo(y,x) ∧ FeelProud(y,x) → evaluation(unexpected))	1	
204	it's expected that you won't cut off family members.	∀x∀**y(FamilyMember(x,y) ∧ ¬CutOff(x,y) → evaluation(expected))**	∀x(**FamilyMember(x) → ¬CutOff(x)**)	1	"For all x: If x is Family member, then x is not Cut off"	∀x∀y(FamilyMember(x,y) → CutOff(y,x) → evaluation(expected))	1	
205	it's bad to show one's feelings for your partner.	∀x∀**y(Partner(x,y) ∧ ShowFeelings(x,y) → evaluation(bad))**	∀x∀y(Partner(x,y) ∧ ShowsFeelings(x,y) → **¬evaluation(good)**)	1	"For all x, for all y: If x is Partner of y and x Show feelings for y, then evaluation is not good"	∀x∀y(Partner(x,y) ∧ ShowsFeelings(x,y) → evaluation(bad))	1	
206	it is good to be the bigger person and not hold a grudge.	∀x∀**y(BiggerPerson(x) ∧ ¬HoldGrudge(x) → evaluation(good))**	∀x∀y(BeBiggerPerson(x) ∧ ¬HoldGrudge(x) → evaluation(good))	1	"For all x, for al y: If x Be bigger person and x don't hold grudge, then evaluation is good"	∀x(BeBiggerPerson(x) ∧ ¬HoldGrudge(x) → evaluation(good))	1	
207	it's not admirable when people are able to admit their mistakes.	∀**x(AbleToAdmitMistakes(x) → ¬Admirable(x))**	∀x(People(x) ∧ AbleToAdmitMistakes(x) → ¬evaluation(admirable))	1		∀x(People(x) ∧ AbleToAdmitMistakes(x) → ¬evaluation(admirable))	1	
208	it's good to care about your family's history.	∀x∀**y(FamilyHistory(y) ∧ Cares(x,y) → evaluation(good))**	∀x(**Family(x) → (CareAboutHistory(x)** → evaluation(good)))	1	"For all x: If x is Family, then (x Care about history, then evaluation is good"	∀x∀y(**Family(x,y) ∧ History(x) → (CareAbout(y,x)** → evaluation(good)))	1	"For all x, for all y: If x is Family of y and x is History, then (y Care about x, then evaluation is good)"
209	it's bad to phrase something awkwardly.	∀x∀**y(IsSomething(y) ∧ PhraseAkwardly(x,y) → evaluation(bad))**	∀x(AwkwardPhrase(x) → **¬evaluation(good)**)	1	"For all x: If x is Awkward phrase, then evaluation is not good"	∀x(PhraseSomething(x) ∧ Awkward(x) → evaluation(bad))	1	"For all x: If x Phrase something and x is Awkward, then evaluation is bad"
210	it's not okay to expect someone care about to be a perfect person.	∀x∀**y(CareAbout(x,y) ∧ ExpectsToBePerfect(x,y) → ¬evaluation(okay))**	∀x∀y(ExpectsPerfectPerson(x,y) → ¬evaluation(ok))	1	"For all x, for all y: If x Expects perfect person y, then evaluation is not okay"	∀x∀y((Person(x) ∧ CareAbout(y,x) ∧ ExpectsPerfectPerson(y,x)) → ¬evaluation(ok))	1	