Predicate Logic Translation Basics + Do’s & Don’ts

* Ontology: propositional logic + +
* Key: the predicates of the sentence translated into predicate logic language
* Write down the predicate’s translated form and indicate what it is supposed to mean
* Ex. Bx = to be bitter
* Domain: the set in which the variable of the predicate belongs to
* If the domain of different variables differ, then indicate them all separately
* Depending on the domain, it can affect the number of predicates (keys) that sentences holds. For example, if the sentence is “all people are smart” and the domain of x is everything, then the translation would be “(Vx)[Px -> Sx]”, but if the domain is already people, then the translation is “(Vx)(Sx)”.
* Proper names (=individual constants): when there is one, certain subject the predicate refers to, that thing/person is translated as a lower case a – v
* Ex. Theodore, you, this man, Mary’s new bike (when it refers to the whole) etc
* Predicate: is the property attributed to the subject/object. It is translated to upper case A-Z
* One place predicate: ex. Mx (to be mortal) / two place predicate: Lxy ( x loves y) / three place predicate: Gxyz (x gets y from z)
* Function?: predicate + individual variable. This refers to a particular property which can be attributed, but alone, it is not a proposition (no truth value).
* Quantifier: either universal Vx or existential Ex.
* All or No: Vx = for all given x, ---
* Some: Ex = for at least one x, --- / for some x, ---
* Brackets: designates the order of interpretation. What in front of a bracket, that thing designates the entire formula behind itself.
* Exemplified in exercise 7 & 8

Also, Vergauwen wants us toput brackets around even single predicates / quantifiers.

* How to make propositions in predicate logic
1. substitution: replace the individual variable with the individual constant (proper name)

ex. Mx = to be mortal. S = Socrates. 🡪 Ms = Socrates is mortal

1. binding: bind the predicate with the quantifier

ex. Mx = to be mortal 🡪 (Vx)(Mx) = for all given x, x is mortal.

Exercises

First of all, interpret the meaning & intention of sentence. 숨겨졌거나 은연 중에 깔려있는 것들 명백하게 하기. 특히, “someone ---“ – if it is a maxim, then it applies to UNIVERSALLY!

* Either it’s ambiguous
* Or it looks existential but is actually universal because it is a maxim

Predicate logic is to make the representation simpler – not all words are going to be literally translated. Ex. but 🡪 mere conjunction!

Specify as much as possible – take the domain as wide as possible and specify all the predicates

Don’t do x is domain of humans, say x is everything and Hx is to be human.

Be careful where you put the quantifier: only put in the forefront if it applies to the whole proposition. If not, apply only to the predicate.

I think the difference between connecting predicates by ^ and just putting ()() is that the former is conjunction, and the latter is of “binding”, so the front bracket applies to the rest of the formula behind it.

Translating no, all, any …

No z that is P1 is P2 = (Vz)(P1z -> ~P2z) Every z that is P1 is not P2

 = ~(Ez)(P1z ^ P2z) It is not the case that some z is P1 and P2

The usual interpretation of any is existential(“at least one”).

Ex. No woman loves any man = no woman loves at least one man = all women do not love any man at all, not even one)

(another possibility exists though: No woman loves every man, meaning that no woman loves all the men in the world)

\*when and when not t specify humans?

\*conjunction = ^ / Vx = universal quantifier / Ex = existential quantifier / implication = 🡪 / negation = ~ / v = or (weak disjunction)

1. Everything is bitter or sweet.
2. Either everything is bitter or everything is sweet.
3. A whale is a mammal

\*interpretation: “A whale” is an abstraction of the category of whales. Therefore, this requires a universal quantifier.

1. Theodore is a whale
2. Mary Ann has a new bicycle
3. This man owns a big car check
* Interpretation: this man = name
* t = this man / Oxy = x owns y / By = y is big / Cy = y is a car
* translation = (Ey)(By ^ Cy ^ Oty)
1. Everybody loves somebody
* Domain: humans
* interpretation: ambiguous
* Lxy = x loves y
* Translation (order of brackets!)
1. there is one person who is loved by everybody = (Ey)(Vx)(Lxy)

(for some y, all x loves that y)

1. everybody has at least one person that they each love = (Vx)(Ey)(Lxy)

(for all given x, x loves some y)

* do I have to specify that x is human as a predicate or is it just fine by domains?
1. There is somebody who is loved by everyone check

Translation (1) from above.

1. Elsie did not get anything from Charles

- name: Elsie = e / Charles = c

- Gxyz = x gets y from z

- translation = (Vy)~(Geyc) = ~(Ey)(Geyc)

1. Lynn gets some present from John, but she doesn’t get anything from Peter

-Lynn = l

-John = j

-Peter = p

-Gxyz = x gets y from z

-Py = y is a present

-Ty = y is a thing

-translation: (Ey)(Py ^ Glyj) ^ (Ey)(Ty ^ ~Glyp)

1. Somebody stole or borrowed Mary’s new bike
* Sxy = x stole y
* Bxy = x borrowed y
* Mary’s new bike = m
* Translation: (Ex)(Sxm v Bxm)
1. You have eaten all my cookies
* Exy = x has eaten y
* Cy = y is a cookie
* My = y is mine
* a = You
* Translation = [(Vy)(Cy ^ My) 🡪 Eay ]
1. Nobody is loved by no one check
* ? can I apply the translation for “no” above for “no one”, or is the meaning different here?
* Lxy = x is loved by y
* Translation = all people are loved by at least one person ??

 = (Vx)(Ey)(Lxy)

 = ~[(Ex)(Ey)~(Lxy)]

1. If all logicians are smart, then Alfred is smart too
* Interpretation: if-then material implication. “too” can be ignored.
* Lx = x is a logician
* Sx = x is smart
* Alfred = a
* Translation: [(Vx)(Lx ^ Sx) 🡪 Sa]
1. Some men and women are not mature check
* M1x = x is a man
* Wx = x is a woman
* M2x = x is mature
* Translation: (Ex)[(M1x ^ ~M2x) ^ (Wx ^ ~M2x)]
1. Barking dogs don’t bite

- B1x = x barks

- Dx = x is a dog

- B2x = x bites

- Translation = (Vx)[(Dx ^ B1x) 🡪 ~(B2x)]

1. If John owns a dog, he has never shown it to anyone check

-never shown it to anyone = is not the case that it has been shown to at least one person

-John = j

-Dy = y is a dog

-Sxyz = x has shown y to z

-Oxy =x owns y

-translation: (Ey)[(Dy ^ Ojy) 🡪 ~(Ez)(Sjyz)]

1. Harry has a beautiful wife, but she hates him check
* But -> conjuction
* Harry = h
* By = y is beautiful
* Wy = y is a wife
* H1xy = x has y
* H2yx = y hates x
* Translation: (Ey)[(By) ^ (Wy) ^ (H1hy) ^ (H2yh)]
1. Nobody lives in Urk who wasn’t born there check

-interpretation: everyone who lives in urk was born there

-Lxy = x lives in y

-Urk = u

-Bxy = x was born in y

-translation: (Vx)[(Lxu) 🡪 ~~(Bxu)] =(Vx)[Lxu 🡪 Bxu]

 = ~[(Ex)(Lxu ^ ~Bxu)]

1. John borrowed a book from Peter but hasn’t given it back to him
* Bxyz = x borrowed y from z
* John = j
* Peter = p
* Gxyz = x give y back to z
* By = y is a book
* Translation: (Ey)[(By ^ Bjyp) ^ ~(Gjyp)]
1. Some people are nice to their bosses even though they are offended by them check
* Though ? / Their?
* Nxy = x is nice to y
* Bxy = y is x’s boss
* Oxy = x is offended by y
* Translation: (Ex)[(Vy){(Bxy) 🡪 (Oxy ^ Nxy)}]
1. Someone who promises something to somebody should do it check
* Interpretation: some”thing” should be “done”/ is a maxim 🡪 refers to everyone
* I’m confused which one is existential and which is universal
* Pxyz = x promises y to z
* Dxy = x should do y
* Ty = y is a thing that can be done
* Hx = x is a human
* Domain: because humans and things are mixed in this world, should have predicate for being human!
* Translation: (Vx)(Ey)(Ez)[(Hx ^ Ty ^ Hz) ^ (Pxyz 🡪Dxy)] 이 구조 기억하기! 전제가 되는 것들은 따로 묶고 실제 행동을 표현하는 predicate은 뒤에, ^으로 묶어두기!
1. People who live in Amherst or close by own a car check
* Px = x is a person
* Amherst = a
* Lxy = x lives in y
* Bxy = x lives close by y
* Oxz = x owns z
* Cz = z is a car
* Translation: (Vx)(Px)[(Lxa >--< Bxa) 🡪 (Ez)(Cz ^ Oxz)]
1. If you see anyone, you should give no letter to her check
* ?? key 설정 어떻게
* Sxz = x sees z
* You = u
* Ly = y is a letter
* Her = h
* Gxyz = x should give y to z
* Translation: (Ez)(Suz) 🡪 {(Vy)[(Ly) 🡪~Guyh]}
1. If Pedro owns donkeys, he beats them
* Pedro = p
* Oxy = x owns y
* Dy = y is a donkey
* Bxy = x beats y
* Translation: (Vy)[(Dy ^ Opy) 🡪 Bpy]
1. Someone who owns no car does own a motorbike check
* Interpretation: does someone here mean everyone?
* Owns no car = does not own a car
* Oxy = x owns y
* Cy = y is a car
* My = M is a motorbike
* (Vx)[(Vy)(Cy ^ ~Oxy) 🡪 (Ey)(My ^ Oxy)]
1. If someone who cannot make a move has lost, then I have lost
* Interpretation: someone?
* Mx = x can make a move
* Lx = x has lost
* I = i
* Translation: (Vx)(~Mx ^ Lx) 🡪 Li
1. Someone has borrowed a motorbike and is riding it
* Bxy = x has borrowed y
* My = y is a motorbike
* Rxy = x is riding y
* Translation: (Ex)(Ey)(My ^ Bxy ^ Rxy)
1. Someone has borrowed a motorbike from somebody and didn’t return it to her
* Bxyz = x has borrowed y from z
* Rxyz = x returns y to z
* My = y is a motorbike
* Translation: (Ex)(Ey)(Ez)(My ^Bxyz ^ ~Rxyz)
1. If someone is noisy, everybody is annoyed

-Nx = x is noisy

-Ay = y is annoyed

-translation: (Ex)(Nx) 🡪 (Vy)(Ay)

-domain of x & y = people

1. If someone is noisy, everybody is annoyed at him

-Ayx = y is annoyed at x

-translation: (Ex)[(Nx) 🡪 (Vy)(Ayx)]

22. Someone who promises something to somebody should do it.

- interpretation: says “some” BUT this is a general maxim which is to apply to everyone. Therefore, is a universal! It means “whoever does this---”.

- key: Pxyz = x promises y to z, Dy = to do y

- domain: x, z = humans / y = things

-translation: (Vx)(Vy)(Ez)[Pxyz -> Dy]

VxVy[(Hx ^ Ay) ^ Ez(Pxyz -> Dxy)] – is the correct one!

H to be human

A to be an act

Mine is too arbitrary. Antonio says that I have to specify as much as possible. I have to have predicates for “being human” and “something is an act”

I thought if the domains specify for me, I wouldn’t have to, but according to him, the domains are supposed to be redundant!

Modal logic is an extension of propositional logic.

Two operators: