lecture 19, the formal language GPL

phil1012 introductory logic

overview

this lecture

- an introduction to the formal language GPL
- the limitations of MPL and the motivation for GPL
- the syntax of GPL
- issues in translation with respect to GPL

learning outcomes

- after doing the relevant reading for this lecture, listening to the lecture, and attending the relevant tutorial, you will be able to:
  - explain what GPL can do that MPL cannot
  - identify well-formed formulas of GPL
  - translate propositions and arguments from English into GPL

required reading

- sections 12.1 and 12.2 of chapter 12

the limitations of MPL

the limitations of MPL

- MPL allows us to attribute properties to individuals. e.g.
  - John is tall
  - Jane is fast
- but it does not allow us to express *relations* between individuals. e.g.
  - John likes Jane
  - Jane does not like John
  - Jane prefers Mark to John

- consider a proposition like this:
  - Bill likes Ben
- we might try to translate it using a glossary like this:
  - Lx: x likes
  - a: Bill
  - b: Ben
- and a translation like this:
  - $(La \wedge Lb)$
- but this doesn't say what we want it to say.

- the predicate 'likes' is a two-place predicate: it requires two names to make a proposition.
- to get from MPL to GPL we just add two- and in general n-place predicates.

- to translate a proposition like this:
  - Bill likes Ben
- we need a glossary like this:
  - L^2xy: x likes y
  - a: Bill
  - b: Ben
- and a translation like this:
  - L^2ab

- to translate a proposition like this:
  - Ben lives in London
- we need a glossary like this:
  - L^2xy: x lives in y
  - b: Ben
  - l: London
- and a translation like this:
  - L^2bl

- to translate a proposition like this:
  - Bill likes someone who lives in London
- we need a glossary like this:
  - N^2xy: x lives in y
  - L^2xy: x loves y
  - P^1x: x is a person
  - a: Bill
  - l: London
- and a translation like this:
  - $\exists y\Big(L^2by \wedge P^1y \wedge N^2yl\Big)$

- to translate a proposition like this:
  - Bill does not like everyone who lives in London
- we need a glossary like this:
  - N^2xy: x lives in y
  - L^2xy: x loves y
  - P^1x: x is a person
  - b: Bill
  - l: London
- and a translation like this:
  - $\neg\forall y\Big(\Big(N^2yl \wedge P^1y\Big) \rightarrow L^2by\Big)$

predicates in GPL

- predicates in GPL are capital letters with a superscript indicating the number of places.
  - $A^1, B^1, C^1, ..., A^2, B^2, C^2, ...$
- we leave off the superscript when no confusion will result.
- we do not use I^2 as a two-place predicate. We reserve it for a special purpose. (It is the 'I' in GPLI.)

the syntax of GPL

- the syntax of GPL is just like the syntax for MPL except for the clause for atomic wffs which now looks like this.

1. Wffs of PL are defined as follows:

1. Where $\underline{P}^n$ is any n-place predicate and $\underline{t}_1 \ldots \underline{t}_n$ are any terms, the following is a wff:

$$\underline{P}^n \underline{t}_1 \ldots \underline{t}_2$$

- that is, an n-place predicate followed by any mixture of n names and/or variables is a well-formed formula.
- wffs of this form are atomic.

order matters

- suppose you want to translate this:
  - Bill is heavier than Mary
- and your glossary looks like this:
  - H^2xy: x is heavier than y
  - b: Bill
  - m: Mary
- then your translation must look like this:
  - H^2bm
- and not like this:
  - H^2mb
- order matters!

order matters

- suppose you want to translate this:
  - Bill is heavier than Mary
- and your glossary looks like this:
  - H^2yx: x is heavier than y
  - b: Bill
  - m: Mary
- then your translation must look like this:
  - H^2mb
- and not like this:
  - H^2bm

- the best way to get a feel for translations into GPL is to look at some examples

---

- let's translate this into GPL:

P1.	Bill is heavier than Mary
P2.	Mary is heavy
C1.	Bill is heavy

- our glossary:
  - H^2yx: y is heavier than x
  - H^1x: x is heavy
  - b: Bill
  - m: Mary
- our tranlation:
  - $H^2bm, H^1m, \therefore H^1b$

- let's translate this into GPL:
  - Singapore is between Sydney and London
- our glossary:

    - B^3xy: x is between y and z
    - g: Singapore
    - s: Sydney
    - l: London
- our translation:
    - B^3gsl

- let's translate this into GPL:
    - Alfred can solve every puzzle.
- our glossary:
    - P^1x: x is a puzzle
    - S^2xy: x can solve y
    - a: Alfred
- our translation:
    - $\forall x(\ Px \rightarrow Sax)$

- let's translate this into GPL:
    - Alfred can solve any puzzle
- our glossary:
    - P^1x: x is a puzzle
    - S^2xy: x can solve y
    - a: Alfred
- our translation:
    - $\forall x(\ Px \rightarrow Sax)$

- let's translate this into GPL:
    - Alfred cannot solve every puzzle.
- our glossary:
    - P^1x: x is a puzzle
    - S^2xy: x can solve y
    - a: Alfred
- our translation:
    - $\neg\forall x(\ Px \rightarrow Sax)$
    - $\exists x(\ Px \wedge \neg Sax)$

- let's translate this into GPL:
    - Alfred cannot solve any puzzle.
- our glossary:
    - P^1x: x is a puzzle
    - S^2xy: x can solve y
    - a: Alfred
- our translation:
    - $\forall x(\ Px \rightarrow \neg Sax)$
    - $\neg\exists x(\ Px \wedge Sax)$

multiple quantifiers in GPL

muliple quantifiers in GPL

- let's take a close look at GPL formulas with multiple quantifiers

---

- consider the open atomic wff Sxy.
- suppose we have the following glossary:
    - Sxy: x sees y

- to make a proposition (a closed wff) from Sxy we must add two quantifiers: one containing x and one containing y.
- they can be existential or universal.

- here are all the possible combinations:

1. $\forall x \forall y Sxy$
2. $\forall y \forall x Sxy$
3. $\exists x \exists y Sxy$
4. $\exists y \exists x Sxy$
5. $\forall x \exists y Sxy$
6. $\exists y \forall x Sxy$
7. $\forall y \exists x Sxy$
8. $\exists x \forall y Sxy$

- what do each of them mean?

- consider:
    - $\forall x \forall y Sxy$
- roughly: for every x, and for every y, x sees y.
- dynamically: no matter what you pick first—call it x—and no matter what you pick second—call it y—x sees y.
- translates: everything sees everything.

- consider:
    - $\forall y \forall x Sxy$
- roughly: for every y, and for every x, x sees y.
- dynamically: No matter what you pick first—call it y—and no matter what you pick second—call it x—x sees y.
- translates: everything sees everything.
- 1 and 2 are equivalent

- consider:
    - $\exists x \exists y Sxy$
- roughly: for some x, and for some y, x sees y.
- dynamically: You can pick a thing-call it x—and then pick a thing—call it y—such that x sees y.
- translates: something sees something.

- consider:
    - $\exists y \exists x Sxy$
- roughly: for some y, and for some x, x sees y.
- dynamically: You can pick a thing-call it y—and then pick a thing—call it x—such that x sees y.
- translates: something sees something.
- 3 and 4 are equivalent.

- consider:
    - $\forall x \exists y Sxy$
- roughly: for all x, and for some y, x sees y.
- dynamically: No matter what you pick first—call it x—you can pick a thing—call it y—such that x sees y.
- translates: everything sees something (not "everything sees something *other than itself*")

- consider:
  - $\exists y \forall x S x y$
- roughly: for some y, and for all x, x sees y
- dynamically: You can pick a thing-call it y—such that no matter what you pick second—call it x—x sees y.
- translates: something is seen by everything
- 5 and 6 are not equivalent

- consider:
  - $\forall y \exists x S x y$
- roughly: for all y, and for some x, x sees y.
- dynamically: No matter what you pick first—call it y—you can pick a thing—call it x—such that x sees y.
- translates: everything is seen by something.

- consider:
  - $\exists x \forall y S x y$
- roughly: for all y, and for some x, x sees y
- dynamically: you can pick a thing-call it x—such that no matter what you pick second—call it y—x sees y.
- translates: something sees everything.

- considering some more examples may be helpful at this stage

---

- let's the following into GPL:
  - Everyone has a father.
- our glossary:
  - Px: x is a person
  - Fxy: x is a father of y
- our translation:
  - $\forall x(\ Px \rightarrow \exists y F y x)$

- let's translate the following into GPL:
  - There is someone who is everyone's father.
- our glossary:
  - Px: x is a person
  - Fxy: x is a father of y
- our translation:
  - $\exists x\big(\ Px \wedge \forall y(\ Py \rightarrow Fxy)\ \big)$

- let's translate the following into GPL:
  - No one lacks a father but not everyone is a father.
- our glossary:
  - Px: x is a person
  - Fxy: x is a father of y
- our translation:
  - $\neg \exists x(\ Px \wedge \neg \exists y F y x)\ \wedge \neg \forall x(\ Px \rightarrow \exists y F x y)$

- let's translate the following into GPL:
  - There is no such thing as a hotel that has no rooms.
- our glossary:
  - Hx: x is a hotel
  - Rx: x is a room
  - Hxy: x has y
- our translation:

- $\neg\exists x(\ Hx \wedge \neg\exists y(\ Ry \wedge Hxy)\ )$

- let's translate the following into GPL:
  - A teacher who assigns a problem that has no solution has no students who like her.
- our glossary:
  - Px: x is a problem
  - Tx: x is a teacher
  - Lxy: x is a solution of y
  - Axy: x assigns y
  - Sxy: x is a student of y
  - Kxy: x likes y
- our translation:
  - $\forall x\forall y(\ (\ Tx \wedge Py \wedge Axy \wedge \neg\exists zLzy)\ \rightarrow \neg\exists w(\ Swx \wedge Kwx)\ )$

wrapping up

this lecture

- the limitations of MPL and the motivation for GPL
- the syntax of GPL
- issues in translation with respect to GPL

next lecture

- lecture 20, the semantics of GPL and trees for GPL