Math 251, Mon 14-Sep-2020 -- Mon 14-Sep-2020
Discrete Mathematics
Fall 2020

Monday, September 14th 2020

Wk 3, Mo
Topic:: Nested quantifiers
Read:: Rosen 1.5
HW:: PS04 due Fri.

$\forall x \exists y$ (y is a soul mate of x)

In Section 1.3, go over Numbers 12 and 62(?)

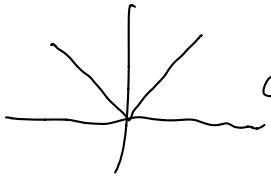
$\exists x \forall y$ (y is a soul mate of x)

Nested quantifiers: like nested loops

Counterexample: how it disproves a universally-quantified statement

→ Every function differentiable at x=a is continuous at x=a. Thm. in Calculus

Every function continuous at x=a is differentiable at x=a. converse



Counterexample to the converse

```
for i = 1:10
   for j = 5:10
      (          )
   end
end
```

Nested quantifiers

- There is a supervisor who oversees every process in this factory
- Every process in this factory is overseen by some person.
- There is a positive integer that is smallest (i.e., at least as small as any other).
- There is a real number that has no reciprocal (multiplicative inverse).
- There is no smallest real number. [Try to write this both with and without a negation symbol.]
- Any secret any person knows can be revealed to the right person.

Rewrite the statement in as simple an English statement as possible. Then write the negation of that statement.

- $\forall$ colors C, $\exists$ an animal A such that A is colored C.
- $\exists$ a book b such that $\forall$ people p, p has read b.
- $\forall$ odd integers n, $\exists$ an integer k such that $n = 2k + 1$.
- ~~$\exists$~~ real x such that for ~~all real~~ y, $x + y = 0$.

Negating nested quantifiers

- $\neg\forall x\exists y P(x, y) \equiv \exists x\forall y\neg P(x, y)$
- $\neg\exists x\forall y P(x, y) \equiv \forall x\exists y\neg P(x, y)$
- $\neg\forall x\forall y P(x, y) \equiv \exists x\exists y\neg P(x, y)$
- $\neg\exists x\exists y P(x, y) \equiv$?
- $\neg\forall x\exists y\forall z P(x, y, z) \equiv$?

ProLog

Basic structures

- names
  - valid characters in names include most any on a keyboard; a name can even contain a space, if the entire name is enclosed in quotation marks.
  - names *beginning* with an upper-case letter or an underscore are reserved for variables
  - *atoms* (or constants) include words beginning with a lower-case letters used in isolation.
  - *predicates* (procedures) include words beginning with a lower-case letters and requiring (as indicated by parentheses) inputs.
- **Conditionals**: if p then q is coded as $q :- p$.
- **Conjunctions**: $p \wedge q$ is coded as p, q.
  Together, the code
  $$r :- p, q$$
  means $p \wedge q \rightarrow r$.
- **Disjunctions**: $p \vee q$ is coded as $p; q$.
- **Negation**: $\neg p$ is coded as $\backslash + p$, meaning the non-negated statement is not provable.

Other important items:

- *Program* files, often referred to as "knowledge bases", can be created separately using a text editor.
- Commands in prolog end with a period (.) character.
- Sample rules:

  `husband(luke, mara) :- wife(mara, luke).`
  expresses wife(mara,luke) → husband(luke,mara)
  modus ponens

  `ownsLightSaber(X) :- jedi(X).`
  expresses $\forall X(\text{jedi}(X) \rightarrow \text{ownsLightSaber}(X))$

  `husband(X, Y) :- wife(Y, X).`
  expresses $\forall X \forall Y(\text{wife}(Y, X) \rightarrow \text{husband}(X, Y))$

- Sample queries:

  `jedi(dooku).`
  evaluates as True since this is a fact in the knowledge base.

  `master(sidious, X).`
  expresses "Is $\exists X$ for which master(sidious,X) is true?" All instances are listed.

Practice

Websites: `http://scofield.site/courses/m251/worksheets/sWars.txt`

`https://www.tutorialspoint.com/execute_prolog_online.php`

1. Write queries for
   (a) whether luke is a child of leia.
   (b) all children of leia.
   (c) all sons of leia.
   (d) all uncles of jacen.
   (e) all grandchildren of anakin
   (f) all names of "force-sensitive" characters (whether sith or jedi)
   (g) all names of characters who are both sith and jedi
2. Write rules for
   (a) mother(X), so that the mother of X is sought/found
   (b) nephew(X)
   (c) isForceSensitive(X)
   (d) isForceSensitive(X)
   (e) grandfather(X)
3. Add information to the knowledge base so that there is a person named owen who appears in response to the query `uncle(luke)`.