



Lecture 3

(A) Predicate logic

(B) Linux

Slides by David Albrecht (2011) and Graham Farr (2013-2014, 2017).

FIT2014 Theory of Computation

Overview

Predicate logic

- Sentences
- Quantifiers
- Knowledge Representation

Linux

Predicate logic: Example

All men are mortal

Socrates is a man.

Therefore Socrates is mortal.

- Objects: **socrates, set of people.**
- Properties: **man, mortal**

Example

There is an app which is loved by every student.

Therefore every student loves some app.

- Objects: **set of people**.
- Properties: **app, student**.
- Relation: **loves**

Objects

- **Constant symbols**
 - Names which refer to exactly one object.
 - socrates, wumpus, I, 2, ...
- **Function symbols**
 - relates some objects to exactly one object.
 - motherOf, kingOf, plus, times, ...
 - Complex name.
- **Individual variables**
 - a variable which can refer to any object.
 - X, Y, ...

Term

A **term** is a logical expression which refers to an object.

E.g.

- **Constant symbols.**
- **Individual variables.**
- **Functions of constant symbols.**
- **Functions of other terms.**

Equality symbol (=)

Used to state that two objects are the same.

rebecca = rebecca

fatherOf(john) = henry

X = kingOf(sweden)

Predicates

- E.g.: man, mortal, app, student, loves, ...
- Properties (1 place)
- Relations (2 or more places)
- Can only be **True** or **False** (like *propositions*)
- they take **arguments** (unlike *propositions*)

Sentences

- **Atomic sentences**
 - A predicate symbol followed by a list of terms in brackets.
 - E.g.
taller(motherOf(claire), mary)
- **Complex sentences**
 - Atomic sentences joined together by logical connectives
 - E.g.
man(socrates) \Rightarrow mortal(socrates)

Universal Quantification

- Used to make a statement about **every** object.
- \forall “**for all**”

All dogs are happy

$$\forall X (\text{dog}(X) \Rightarrow \text{happy}(X))$$

No dog is happy

All dogs are unhappy

$$\forall X (\text{dog}(X) \Rightarrow \neg \text{happy}(X))$$

Existential Quantification

- Used to make a statement about **some** object.
- \exists “**there exists**”

Some dogs are happy

$$\exists X (\text{dog}(X) \wedge \text{happy}(X))$$

Some dogs are not happy

$$\exists X (\text{dog}(X) \wedge \neg \text{happy}(X))$$

Universe of Discourse

- The set of objects that are being referred to.
 - Often it is unstated or assumed.
 - Can affect the truth of a statement.
-
- Consider the predicate **greaterThanZero**.

$$\forall X \text{greaterThanZero}(X)$$

Doing logic with quantifiers

If we know that

$$\forall X \text{ blah}(X)$$

and **obj** is any specific object

(in the universe of discourse),
then we can deduce that

$$\text{blah}(\text{obj})$$

We have:

$$(\forall X \text{ blah}(X)) \Rightarrow \text{blah}(\text{obj})$$

Also:

$$\text{blah}(\text{obj}) \Rightarrow (\exists X \text{ blah}(X))$$

Doing logic with quantifiers

$$\forall X (p(X) \wedge q(X))$$

is logically equivalent to

$$(\forall X p(X)) \wedge (\forall X q(X))$$

$$\exists X (p(X) \vee q(X))$$

is logically equivalent to

$$(\exists X p(X)) \vee (\exists X q(X))$$

What about the logical relationship between ...

$$\forall X (p(X) \vee q(X))$$

and $(\forall X p(X)) \vee (\forall X q(X))$...?

... etc

Relationship between quantifiers

$\neg \forall y$ means the same as $\exists y \neg$

Not all dogs are happy.

is the same as ... There exists an unhappy dog.

$$\neg \forall X (\text{dog}(X) \Rightarrow \text{happy}(X))$$

Not all dogs are happy

$$= \exists X \neg (\text{dog}(X) \Rightarrow \text{happy}(X))$$

$$= \exists X \neg (\neg \text{dog}(X) \vee \text{happy}(X))$$

(see last lecture)

$$= \exists X (\neg \text{dog}(X) \wedge \neg \text{happy}(X))$$

(by De Morgan)

$$= \exists X (\text{dog}(X) \wedge \neg \text{happy}(X))$$

There exists an unhappy dog

Relationship between quantifiers

Similarly,

$\neg \exists y$ means the same as $\forall y \neg$

$\neg \forall y \neg$ means the same as

$\neg \exists y \neg$ means the same as

Socrates Example

- All men are mortal

$$\forall X (\text{man}(X) \Rightarrow \text{mortal}(X))$$

- Socrates is a man.

man(socrates)

- Socrates is mortal.

mortal(socrates)

Love Example

- ▶ There is an app which is loved by every student.
- ▶ There is an app X and if Y is a student then Y loves it.
 $\exists X (\text{app}(X) \wedge \forall Y (\text{student}(Y) \Rightarrow \text{loves}(Y, X)))$

- ▶ Every student loves some app.
- ▶ For every student Y there exists an app X that she loves.
 $\forall Y (\text{student}(Y) \Rightarrow \exists X (\text{app}(X) \wedge \text{loves}(Y, X)))$

Unix

Origin:

- Bell Laboratories, 1969; first published in 1974
- Dennis M. Ritchie and Ken Thompson
- cost-effective, simple, elegant, easy to use
- widely used, e.g. for servers
- security



Unix

- Dennis M. Ritchie and Ken Thompson, The Unix Time-Sharing System, *Communications of the ACM* 17 (no. 7) (July 1974) 365-375.
- <https://people.eecs.berkeley.edu/~brewer/cs262/unix.pdf>
- From the Abstract:
 - “It offers a number of features seldom found even in larger operating systems, including: (1) a hierarchical file system incorporating demountable volumes; (2) compatible file, device, and inter-process I/O; (3) the ability to initiate asynchronous processes; (4) system command language selectable on a per-user basis; and (5) over 100 subsystems including a dozen languages.”
- written in high-level language, C (mostly)
- can combine programs to make more complex ones

GNU

- GNU project
- Richard Stallman, from 1983
- Aim: free software, developed by large-scale collaboration
- including operating system based on Unix
- **GNU = GNU's Not Unix** (recursive acronym!)
- Much software, but lacked kernel ...
- **GNU General Public Licence**
 - <https://www.gnu.org/licenses/gpl.html>



Linux

- Linux kernel:
- Linus Torvalds, 1991
 - (independently of GNU)
- Then released under GNU Public Licence, 1992
- The OS is sometimes called GNU/Linux



Drawing by Pekka Vuori, 2000

MSc thesis, University of Helsinki, 1997:

Linux: A Portable Operating System

https://www.cs.helsinki.fi/u/kutvonen/index_files/linus.pdf