



BITS Pilani
Pilani Campus



CS/IS F214 Logic in Computer Science

MODULE: PREDICATE LOGIC

Predicate Logic – Substitution and Binding

Variables and Substitution

- Variables are placeholders:
 - we may substitute variables with some concrete instances (i.e. *terms*) while interpreting (or understanding) a formula
 - equivalently, we may replace ***a variable node*** in the ***parse tree*** with a ***sub-tree for a term***.



Substitution

- Definition [Substitution]:
 - Given a formula ϕ , a term t , and a variable X ,
 - define $\phi[t/x]$ to be
 - *the formula obtained by replacing each **free occurrence** of variable X in ϕ with t .*
- Before we understand and apply this definition we need to understand
 - free occurrences and bound occurrences of variables.



Binding in Predicate Logic

- A ***binding***, in our context, namely in Predicate Logic refers to
 - *connecting a variable with a part of the formula in which it occurs*
- In Predicate Logic:
 - a binding is achieved by the use of a quantifier – *universal or existential*.



Binding – Generic Definition and Example

- A **binding**, in general, in the context of languages and syntax refers to
 - *connecting a variable (or a symbol) with the text in which it occurs*
- e.g.
 - Consider a program fragment, in a programming language, say **C**
 - **int fact(int n) { return n=0 ? 1 : fact(n-1); }**
 - The variable **n** is bound to the body of function **fact**
 - by being declared a parameter of that function.



Binding – Predicate Logic - Example

- e.g.
 - Consider a predicate logic formula:
 - $\forall X \forall Y p(X) \wedge p(Y) \wedge \neg \text{equals}(X,Y) \rightarrow \forall Z p(Z)$
 - The variables **X** and **Y** are bound
 - to the rest of the formula
 - by way of being quantified in the beginning
 - and the variable **Z** is bound
 - by way of being quantified at the beginning of the right hand side of the implication
 - to the rest of the right hand side of the implication



Bound Variables

- A **bound variable** is a variable that has a binding.
 - In predicate logic:
 - a bound variable is a variable that has been quantified.
- A **bound occurrence** is an occurrence of a variable inside the text to which it is bound
 - In predicate logic:
 - a bound occurrence of a variable is
 - an occurrence inside a formula to which the variable is bound through a quantifier.



ASIDE: Bound Variables and Programming

- Incidentally, if you consider the two following C program fragments

1	<code>int fact(int n) { return n=0 ? 1 : fact(n-1); }</code>
2	<code>const float G; ... float velocityOfDroppedObject(float lapsedT) { return G*lapsedT; }</code>

the symbols `n` and `lapsedT` are universally quantified and `G` is (likely) existentially quantified:

- Why?



Free Variables - Examples

- e.g. In the formula:
 - $\forall X \exists Y p(X) \wedge q(Y) \rightarrow r(Z) \wedge r(Y)$

Z occurs free

- A variable occurring free in a formula is referred to as *a free variable*
 - In the above example, **Z** is a free variable (in the formula given) whereas **X** and **Y** are bound variables.
- But if you consider the sub-formula $r(Z) \wedge r(Y)$ both **Y** and **Z** are free variables.

