Lecture 16: First-order Logic — Prenex Normal Form

Barnaby Martin, 6 March 2021

barnaby.d.martin@durham.ac.uk

Outline

- Logical equivalence (last week)
- Some specific equivalences (last week)
- Prenex normal form (*today*)
- Resolution for first-order logic (next week)

Logical equivalence

Recall that two formulae ϕ and ψ are logically equivalent if they are true for the same set of models, in which case we write

$$\phi \equiv \psi.$$

Parse trees

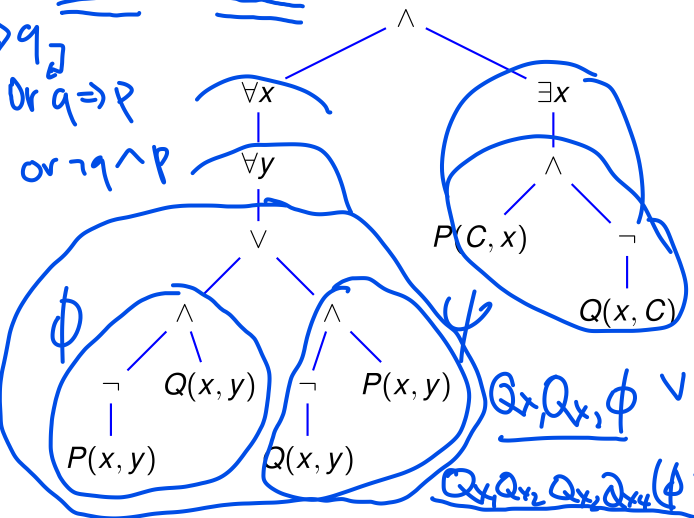
Recall that to check if a formula is well-formed we can use a parse tree. We illustrated this with

$\forall x(\forall y(P(x,y) \Leftrightarrow \neg Q(x,y))) \wedge \exists x(P(C,x) \wedge \neg Q(x,C))$

$p \Leftrightarrow q$

$p \Rightarrow q$ or $q \Rightarrow p$

$\neg p \wedge q$ or $\neg q \wedge p$



$Q_{x_1}Q_{x_2}\phi \vee Q_{x_3}Q_{x_4}\psi$

$Q_{x_1}Q_{x_2}Q_{x_3}Q_{x_4}(\phi \vee \psi)$

Prenex normal form

We are now in a position to obtain an important normal form.
Consider the process of constructing a first-order formula:
We start from atoms and construct increasingly more complex formulae using $\underbrace{\wedge, \vee, \neg,}_{}$ $\exists$, and $\forall$, with the structure of a formula given by its parse tree.

complete

Prenex normal form

We are now in a position to obtain an important normal form. Consider the process of constructing a first-order formula: We start from atoms and construct increasingly more complex formulae using $\wedge$, $\vee$, $\neg$, $\exists$, and $\forall$, with the structure of a formula given by its parse tree.

We say that a first-order formula is in prenex normal form if it is written in the form

$$\forall a \forall b \; \underbrace{P(a,b)}_{\text{unit formula}}$$

$$Q_1 x_1 Q_2 x_2 \ldots Q_k x_k \phi,$$

where:

- each Q_i is a quantifier,
- each x_i is a variable,
- the formula ϕ is quantifier-free.

Prenex normal form

We are now in a position to obtain an important normal form.

Consider the process of constructing a first-order formula:

We start from atoms and construct increasingly more complex formulae using $\wedge$, $\vee$, $\neg$, $\exists$, and $\forall$, with the structure of a formula given by its parse tree.

We say that a first-order formula is in prenex normal form if it is written in the form

$$Q_1 x_1 Q_2 x_2 \ldots Q_k x_k \phi,$$

where:

- each Q_i is a quantifier,
- each x_i is a variable,
- the formula ϕ is quantifier-free.

We shall show that every first-order formula is equivalent to one in prenex normal form.

Establishing prenex normal form

We use the parse tree of a formula ϕ in order to construct an equivalent formula in prenex normal form.

Establishing prenex normal form

We use the parse tree of a formula ϕ in order to construct an equivalent formula in prenex normal form.

A key observation is that if we choose some node of the parse tree of ϕ and look at the sub-tree with this node as root then this sub-tree corresponds to a sub-formula of ϕ; that is, to a formula appearing as a formula within ϕ.

Establishing prenex normal form

We use the parse tree of a formula ϕ in order to construct an equivalent formula in prenex normal form.

A key observation is that if we choose some node of the parse tree of ϕ and look at the sub-tree with this node as root then this sub-tree corresponds to a sub-formula of ϕ; that is, to a formula appearing as a formula within ϕ.

What we do is start at the leaves of the parse tree and work up the tree repeatedly constructing prenex normal form formulae that are equivalent to the formulae corresponding to sub-trees of the parse tree.

Establishing prenex normal form

Let's start at the leaves. The formula at each leaf is trivially in prenex normal form (as it involves no quantifiers).

Establishing prenex normal form

Let's start at the leaves. The formula at each leaf is trivially in prenex normal form (as it involves no quantifiers).

Suppose that we have reached a node of the parse tree that is an $\wedge$-node and that we have constructed prenex normal form formulae equivalent to the formulae corresponding to the sub-trees rooted at the 2 children of this $\wedge$-node.

$$\underbrace{Qx_1 Qx_2 \ldots Qx_n}(\phi'' \wedge \psi'')$$

ϕ'' ψ'' quantifier free version

$$\begin{tikzcd} & \wedge \arrow[dl, dash] \arrow[dr, dash] & \\ \text{(PNF) } \phi' & & \psi' \text{ (PNF)} \end{tikzcd}$$

Establishing prenex normal form

So, the formula corresponding to the sub-tree rooted at this $\wedge$-node is of the form $\psi \wedge \chi$ and we have already constructed ψ' and χ' such that:

- ψ' and χ' are in prenex normal form:
  - ψ' is $Q_1 x_1 Q_2 x_2 \cdots Q_k x_k \psi''$ with ψ'' quantifier-free (and each Q_i a quantifier)
  - χ' is $P_1 y_1 P_2 y_2 \cdots P_k y_k \chi''$ with χ'' quantifier-free (and each P_i a quantifier)
- $\psi \equiv \psi'$
- $\chi \equiv \chi'$

Note that by renaming bound variables (if necessary) we may assume that no x_i is the same variable as any y_j.

Establishing prenex normal form

So, the formula corresponding to the sub-tree rooted at this $\wedge$-node is of the form $\psi \wedge \chi$ and we have already constructed ψ' and χ' such that:

- ψ' and χ' are in prenex normal form:
  - ψ' is $Q_1 x_1 Q_2 x_2 \cdots Q_k x_k \psi''$ with ψ'' quantifier-free (and each Q_i a quantifier)
  - χ' is $P_1 y_1 P_2 y_2 \cdots P_k y_k \chi''$ with χ'' quantifier-free (and each P_i a quantifier)
- $\psi \equiv \psi'$
- $\chi \equiv \chi'$

Note that by renaming bound variables (if necessary) we may assume that no x_i is the same variable as any y_j.
So $\psi \wedge \chi$ is equivalent to a formula in prenex normal form:

$$\begin{aligned} \psi \wedge \chi &\equiv Q_1 x_1 Q_2 x_2 \cdots Q_k x_k \psi'' \wedge P_1 y_1 P_2 y_2 \cdots P_k y_k \chi'' \\ &\equiv Q_1 x_1 Q_2 x_2 \cdots Q_k x_k P_1 y_1 P_2 y_2 \cdots P_k y_k (\psi'' \wedge \chi'') \end{aligned}$$

Establishing prenex normal form

The same construction works for a $\vee$-node of our parse tree.

Establishing prenex normal form

$$\overline{(a \wedge b \wedge c)} = \bar{a} \vee \bar{b} \vee \bar{c}$$

The same construction works for a $\vee$-node of our parse tree.

Consider a $\neg$-node.

Now we have that the formula corresponding to the sub-tree rooted at this $\neg$-node is equivalent to a formula of the form $Q_1 x_1 Q_2 x_2 \cdots Q_k x_k \psi''$ with ψ'' quantifier-free (and each Q_i a quantifier).

$\wedge$ $\vee$

$$\neg \; \psi \qquad \neg(Q_1 x_1 Q_2 x_2 \cdots Q_k x_k \psi'')$$
$$= Q_1' x_1 Q_2' x_2 \cdots Q_k' x_k \neg \psi'' \quad \text{(PNF)}$$

Establishing prenex normal form

The same construction works for a $\vee$-node of our parse tree.

Consider a $\neg$-node.

Now we have that the formula corresponding to the sub-tree rooted at this $\neg$-node is equivalent to a formula of the form $Q_1 x_1 Q_2 x_2 \cdots Q_k x_k \psi''$ with ψ'' quantifier-free (and each Q_i a quantifier).

Hence, using our general rule from earlier, this formula is equivalent to
$Q_1 x_1 Q_2 x_2 \cdots Q_k x_k \neg\psi''$ which is in prenex form.

Establishing prenex normal form

Consider a $\forall$-node.

Now we have that the formula corresponding to the sub-tree rooted at this $\forall$-node is equivalent to a formula of the form: $Q_1 x_1 Q_2 x_2 \cdots Q_k x_k \psi''$ with ψ'' quantifier-free (and each Q_i a quantifier).

$\forall x \phi$ $\quad$ $\forall x \{ \phi : \{ Q_1 x_1 Q_2 x_2 \cdots Q_k x_k \phi'' \} \}$

$\forall x Q_1 x_1 Q_2 x_2 \cdots Q_k x_k \phi''$

Establishing prenex normal form

$\forall x$
$|$
ϕ

$\forall x \phi$

$\forall x Q_1 x_1 Q_2 x_2 \cdots Q_k x_k \phi''$

Consider a $\forall$-node.

Now we have that the formula corresponding to the sub-tree rooted at this $\forall$-node is equivalent to a formula of the form: $Q_1 x_1 Q_2 x_2 \cdots Q_k x_k \psi''$ with ψ'' quantifier-free (and each Q_i a quantifier).

However, this is immediately in prenex normal form (the same construction works for an $\exists$-node of our parse tree).

Hence, continuing in this way yields an equivalent formula to ϕ in prenex normal form.

An illustration

Consider the first-order formula ϕ defined as:

$$\neg\forall x(\exists y \forall z(E(x,y) \vee \neg\forall w \neg P(w,y,z)) \vee \neg\forall y(E(y,x) \wedge \neg M(y)))$$

How is it written in prenex form?

Answer

$$\exists x \forall y \exists z \forall w \forall t \neg((E(x, y) \vee P(w, y, z)) \vee \neg(E(t, x) \wedge \neg M(t)))$$