CS228 : Logic for Computer Science

Module 3: Predicate Logic

Lecture 17: Normal Forms for FOL

Instructor: S. Akshay

IIT Bombay, India

Topics Covered till date

Module 1: Propositional Logic

1. Syntax
2. Semantics
3. Formal proof systems
4. Normal Forms

Module 2: The Problem of Satisfiability

1. Propositional Resolution
2. DPLL
3. CDCL
4. SAT/SMT tool demo

Topics Covered till date

Module 1: Propositional Logic

1. Syntax
2. Semantics
3. Formal proof systems
4. Normal Forms

Module 2: The Problem of Satisfiability

1. Propositional Resolution
2. DPLL
3. CDCL
4. SAT/SMT tool demo

Module 3: Predicate Logic

1. Syntax of predicate logic (Lecture 12)
2. Semantics of predicate logic (Lecture 13)
3. Natural deduction for FOL (Lecture 14)
4. Soundness, completeness, and structures in FOL (Lecture 15)
5. Expressiveness and the limits of FOL (Lecture 16)
6. Today: Normal forms for FOL

Pop Quiz Yet Again!

1. Let m be a constant, f a function symbol of arity 1, and S, B predicate symbols of arity 2. Which of the following are well-formed formulas? Which of them are atomic?

   i. $S(m, x)$
   ii. $B(m, f(m))$
   iii. $f(m)$
   iv. $B(B(m, x), y)$
   v. $S(B(m), z)$
   vi. $(B(x, y) \to (\exists z\, S(z, y)))$
   vii. $(S(x, y) \to S(y, f(f(x))))$
   viii. $(B(x) \to B(B(x)))$

Pop Quiz Yet Again!

1. Let m be a constant, f a function symbol of arity 1, and S, B predicate symbols of arity 2. Which of the following are well-formed formulas? Which of them are atomic?

   i. $S(m, x)$
   ii. $B(m, f(m))$
   iii. $f(m)$
   iv. $B(B(m, x), y)$
   v. $S(B(m), z)$
   vi. $(B(x, y) \to (\exists z\, S(z, y)))$
   vii. $(S(x, y) \to S(y, f(f(x))))$
   viii. $(B(x) \to B(B(x)))$

2. Let Q be a predicate symbol of arity 3, g a function symbol of arity 2, c a constant.
   - $\varphi_1 := \forall x \forall y\, Q(g(x, y), g(y, y), c)$ $\quad$ $\varphi_2 := \forall x \forall y\, Q(g(x, y), g(y, y), z)$
   - $\varphi_3 := \forall x \forall y \forall z\, Q(g(x, y), g(y, y), z)$ $\quad$ $\varphi_4 := \forall x \forall y \exists z\, Q(g(x, y), g(y, y), z)$

   Under $M = (\mathbb{N}, g^M, Q^M, c^M)$, with $g^M(x, y) := 0$ for all x, y; $Q^M := \{(a, b, d) \mid d = 0\}$; $c^M := 0$, what is the truth value of $\varphi_1, \varphi_2, \varphi_3, \varphi_4$?

Pop Quiz Yet Again!

1. Let m be a constant, f a function symbol of arity 1, and S, B predicate symbols of arity 2. Which of the following are well-formed formulas? Which of them are atomic?

   i. $S(m, x)$ — well-formed – atomic
   ii. $B(m, f(m))$
   iii. $f(m)$
   iv. $B(B(m, x), y)$
   v. $S(B(m), z)$
   vi. $(B(x, y) \to (\exists z\, S(z, y)))$
   vii. $(S(x, y) \to S(y, f(f(x))))$
   viii. $(B(x) \to B(B(x)))$

2. Let Q be a predicate symbol of arity 3, g a function symbol of arity 2, c a constant.
   - $\varphi_1 := \forall x \forall y\, Q(g(x, y), g(y, y), c)$ $\quad$ $\varphi_2 := \forall x \forall y\, Q(g(x, y), g(y, y), z)$
   - $\varphi_3 := \forall x \forall y \forall z\, Q(g(x, y), g(y, y), z)$ $\quad$ $\varphi_4 := \forall x \forall y \exists z\, Q(g(x, y), g(y, y), z)$

   Under $M = (\mathbb{N}, g^M, Q^M, c^M)$, with $g^M(x, y) := 0$ for all x, y; $Q^M := \{(a, b, d) \mid d = 0\}$; $c^M := 0$, what is the truth value of $\varphi_1, \varphi_2, \varphi_3, \varphi_4$?

Pop Quiz Yet Again!

1. Let m be a constant, f a function symbol of arity 1, and S, B predicate symbols of arity 2. Which of the following are well-formed formulas? Which of them are atomic?

i.	$S(m, x)$	well-formed – atomic
ii.	$B(m, f(m))$	well-formed – atomic
iii.	$f(m)$	
iv.	$B(B(m, x), y)$	
v.	$S(B(m), z)$	
vi.	$(B(x, y) \to (\exists z\, S(z, y)))$	
vii.	$(S(x, y) \to S(y, f(f(x))))$	
viii.	$(B(x) \to B(B(x)))$	

2. Let Q be a predicate symbol of arity 3, g a function symbol of arity 2, c a constant.
   - $\varphi_1 := \forall x \forall y\, Q(g(x, y), g(y, y), c)$ $\quad$ $\varphi_2 := \forall x \forall y\, Q(g(x, y), g(y, y), z)$
   - $\varphi_3 := \forall x \forall y \forall z\, Q(g(x, y), g(y, y), z)$ $\quad$ $\varphi_4 := \forall x \forall y \exists z\, Q(g(x, y), g(y, y), z)$

   Under $M = (\mathbb{N}, g^M, Q^M, c^M)$, with $g^M(x, y) := 0$ for all x, y; $Q^M := \{(a, b, d) \mid d = 0\}$; $c^M := 0$, what is the truth value of $\varphi_1, \varphi_2, \varphi_3, \varphi_4$?

Pop Quiz Yet Again!

1. Let m be a constant, f a function symbol of arity 1, and S, B predicate symbols of arity 2. Which of the following are well-formed formulas? Which of them are atomic?

i.	$S(m, x)$	well-formed – atomic
ii.	$B(m, f(m))$	well-formed – atomic
iii.	$f(m)$	not a formula (a term)
iv.	$B(B(m, x), y)$	
v.	$S(B(m), z)$	
vi.	$(B(x, y) \to (\exists z\, S(z, y)))$	
vii.	$(S(x, y) \to S(y, f(f(x))))$	
viii.	$(B(x) \to B(B(x)))$	

2. Let Q be a predicate symbol of arity 3, g a function symbol of arity 2, c a constant.
   - $\varphi_1 := \forall x \forall y\, Q(g(x, y), g(y, y), c)$ $\quad$ $\varphi_2 := \forall x \forall y\, Q(g(x, y), g(y, y), z)$
   - $\varphi_3 := \forall x \forall y \forall z\, Q(g(x, y), g(y, y), z)$ $\quad$ $\varphi_4 := \forall x \forall y \exists z\, Q(g(x, y), g(y, y), z)$

   Under $M = (\mathbb{N}, g^M, Q^M, c^M)$, with $g^M(x, y) := 0$ for all x, y; $Q^M := \{(a, b, d) \mid d = 0\}$; $c^M := 0$, what is the truth value of $\varphi_1, \varphi_2, \varphi_3, \varphi_4$?

Pop Quiz Yet Again!

1. Let m be a constant, f a function symbol of arity 1, and S, B predicate symbols of arity 2. Which of the following are well-formed formulas? Which of them are atomic?

i.	$S(m, x)$	well-formed – atomic
ii.	$B(m, f(m))$	well-formed – atomic
iii.	$f(m)$	not a formula (a term)
iv.	$B(B(m, x), y)$	ill-formed ($B(m, x)$ isn't a term)
v.	$S(B(m), z)$	
vi.	$(B(x, y) \to (\exists z\, S(z, y)))$	
vii.	$(S(x, y) \to S(y, f(f(x))))$	
viii.	$(B(x) \to B(B(x)))$	

2. Let Q be a predicate symbol of arity 3, g a function symbol of arity 2, c a constant.
   - $\varphi_1 := \forall x \forall y\, Q(g(x, y), g(y, y), c)$ $\quad$ $\varphi_2 := \forall x \forall y\, Q(g(x, y), g(y, y), z)$
   - $\varphi_3 := \forall x \forall y \forall z\, Q(g(x, y), g(y, y), z)$ $\quad$ $\varphi_4 := \forall x \forall y \exists z\, Q(g(x, y), g(y, y), z)$

   Under $M = (\mathbb{N}, g^M, Q^M, c^M)$, with $g^M(x, y) := 0$ for all x, y; $Q^M := \{(a, b, d) \mid d = 0\}$; $c^M := 0$, what is the truth value of $\varphi_1, \varphi_2, \varphi_3, \varphi_4$?

Pop Quiz Yet Again!

1. Let m be a constant, f a function symbol of arity 1, and S, B predicate symbols of arity 2. Which of the following are well-formed formulas? Which of them are atomic?

i.	$S(m, x)$	well-formed – atomic
ii.	$B(m, f(m))$	well-formed – atomic
iii.	$f(m)$	not a formula (a term)
iv.	$B(B(m, x), y)$	ill-formed ($B(m, x)$ isn't a term)
v.	$S(B(m), z)$	ill-formed (B wrong arity, and not a term)
vi.	$(B(x, y) \to (\exists z\, S(z, y)))$	
vii.	$(S(x, y) \to S(y, f(f(x))))$	
viii.	$(B(x) \to B(B(x)))$	

2. Let Q be a predicate symbol of arity 3, g a function symbol of arity 2, c a constant.
   - $\varphi_1 := \forall x \forall y\, Q(g(x, y), g(y, y), c)$ $\quad$ $\varphi_2 := \forall x \forall y\, Q(g(x, y), g(y, y), z)$
   - $\varphi_3 := \forall x \forall y \forall z\, Q(g(x, y), g(y, y), z)$ $\quad$ $\varphi_4 := \forall x \forall y \exists z\, Q(g(x, y), g(y, y), z)$

   Under $M = (\mathbb{N}, g^M, Q^M, c^M)$, with $g^M(x, y) := 0$ for all x, y; $Q^M := \{(a, b, d) \mid d = 0\}$; $c^M := 0$, what is the truth value of $\varphi_1, \varphi_2, \varphi_3, \varphi_4$?

Pop Quiz Yet Again!

1. Let m be a constant, f a function symbol of arity 1, and S, B predicate symbols of arity 2. Which of the following are well-formed formulas? Which of them are atomic?

i.	$S(m, x)$	well-formed – atomic
ii.	$B(m, f(m))$	well-formed – atomic
iii.	$f(m)$	not a formula (a term)
iv.	$B(B(m, x), y)$	ill-formed ($B(m, x)$ isn't a term)
v.	$S(B(m), z)$	ill-formed (B wrong arity, and not a term)
vi.	$(B(x, y) \to (\exists z\, S(z, y)))$	well-formed – not atomic
vii.	$(S(x, y) \to S(y, f(f(x))))$	
viii.	$(B(x) \to B(B(x)))$	

2. Let Q be a predicate symbol of arity 3, g a function symbol of arity 2, c a constant.
   - $\varphi_1 := \forall x \forall y\, Q(g(x, y), g(y, y), c)$ $\quad$ $\varphi_2 := \forall x \forall y\, Q(g(x, y), g(y, y), z)$
   - $\varphi_3 := \forall x \forall y \forall z\, Q(g(x, y), g(y, y), z)$ $\quad$ $\varphi_4 := \forall x \forall y \exists z\, Q(g(x, y), g(y, y), z)$

   Under $M = (\mathbb{N}, g^M, Q^M, c^M)$, with $g^M(x, y) := 0$ for all x, y; $Q^M := \{(a, b, d) \mid d = 0\}$; $c^M := 0$, what is the truth value of $\varphi_1, \varphi_2, \varphi_3, \varphi_4$?

Pop Quiz Yet Again!

1. Let m be a constant, f a function symbol of arity 1, and S, B predicate symbols of arity 2. Which of the following are well-formed formulas? Which of them are atomic?

i.	$S(m, x)$	well-formed – atomic
ii.	$B(m, f(m))$	well-formed – atomic
iii.	$f(m)$	not a formula (a term)
iv.	$B(B(m, x), y)$	ill-formed ($B(m, x)$ isn't a term)
v.	$S(B(m), z)$	ill-formed (B wrong arity, and not a term)
vi.	$(B(x, y) \to (\exists z\, S(z, y)))$	well-formed – not atomic
vii.	$(S(x, y) \to S(y, f(f(x))))$	well-formed – not atomic
viii.	$(B(x) \to B(B(x)))$	

2. Let Q be a predicate symbol of arity 3, g a function symbol of arity 2, c a constant.
   - $\varphi_1 := \forall x \forall y\, Q(g(x, y), g(y, y), c)$ $\quad$ $\varphi_2 := \forall x \forall y\, Q(g(x, y), g(y, y), z)$
   - $\varphi_3 := \forall x \forall y \forall z\, Q(g(x, y), g(y, y), z)$ $\quad$ $\varphi_4 := \forall x \forall y \exists z\, Q(g(x, y), g(y, y), z)$

   Under $M = (\mathbb{N}, g^M, Q^M, c^M)$, with $g^M(x, y) := 0$ for all x, y; $Q^M := \{(a, b, d) \mid d = 0\}$; $c^M := 0$, what is the truth value of $\varphi_1, \varphi_2, \varphi_3, \varphi_4$?

Pop Quiz Yet Again!

1. Let m be a constant, f a function symbol of arity 1, and S, B predicate symbols of arity 2. Which of the following are well-formed formulas? Which of them are atomic?

i.	$S(m, x)$	well-formed – atomic
ii.	$B(m, f(m))$	well-formed – atomic
iii.	$f(m)$	not a formula (a term)
iv.	$B(B(m, x), y)$	ill-formed ($B(m, x)$ isn't a term)
v.	$S(B(m), z)$	ill-formed (B wrong arity, and not a term)
vi.	$(B(x, y) \to (\exists z\, S(z, y)))$	well-formed – not atomic
vii.	$(S(x, y) \to S(y, f(f(x))))$	well-formed – not atomic
viii.	$(B(x) \to B(B(x)))$	ill-formed (wrong arity)

2. Let Q be a predicate symbol of arity 3, g a function symbol of arity 2, c a constant.
   - $\varphi_1 := \forall x \forall y\, Q(g(x, y), g(y, y), c)$ $\quad$ $\varphi_2 := \forall x \forall y\, Q(g(x, y), g(y, y), z)$
   - $\varphi_3 := \forall x \forall y \forall z\, Q(g(x, y), g(y, y), z)$ $\quad$ $\varphi_4 := \forall x \forall y \exists z\, Q(g(x, y), g(y, y), z)$

   Under $M = (\mathbb{N}, g^M, Q^M, c^M)$, with $g^M(x, y) := 0$ for all x, y; $Q^M := \{(a, b, d) \mid d = 0\}$; $c^M := 0$, what is the truth value of $\varphi_1, \varphi_2, \varphi_3, \varphi_4$?

Pop Quiz Contd.

Note first: $Q(g(x,y), g(y,y), \cdot)$ collapses to "$(\cdot = 0)$", independent of x, y.

(a) φ_1 (constant c, c^M=0): true – M alone decides it, since c^M is already fixed at 0.

Pop Quiz Contd.

Note first: $Q(g(x,y), g(y,y), \cdot)$ collapses to "$(\cdot = 0)$", independent of x, y.

(a) φ_1 (constant c, c^M=0): true – M alone decides it, since c^M is already fixed at 0.

(b) φ_2 (free z): cannot be determined from M alone!
z is free, so we also need an environment α

- e.g. $\alpha(z)$:=0 gives $M, \alpha \models \varphi_2$, but $\alpha'(z)$:=1 gives $M, \alpha' \not\models \varphi_2$. Same M, two different answers.

Pop Quiz Contd.

Note first: $Q(g(x,y), g(y,y), \cdot)$ collapses to "$(\cdot = 0)$", independent of x, y.

(a) φ_1 (constant c, c^M=0): true – M alone decides it, since c^M is already fixed at 0.

(b) φ_2 (free z): cannot be determined from M alone!
z is free, so we also need an environment α

- e.g. $\alpha(z)$:=0 gives $M, \alpha \models \varphi_2$, but $\alpha'(z)$:=1 gives $M, \alpha' \not\models \varphi_2$. Same M, two different answers.

(c) φ_3 (z bound by $\forall$): false – $\forall z\,(z = 0)$ fails in $\mathbb{N}$ (e.g. $z = 1$). M alone decides it.

Pop Quiz Contd.

Note first: $Q(g(x,y), g(y,y), \cdot)$ collapses to "$(\cdot = 0)$", independent of x, y.

(a) φ_1 (constant c, c^M=0): true – M alone decides it, since c^M is already fixed at 0.

(b) φ_2 (free z): cannot be determined from M alone!
z is free, so we also need an environment α

- e.g. $\alpha(z)$:=0 gives $M, \alpha \models \varphi_2$, but $\alpha'(z)$:=1 gives $M, \alpha' \not\models \varphi_2$. Same M, two different answers.

(c) φ_3 (z bound by $\forall$): false – $\forall z\,(z = 0)$ fails in $\mathbb{N}$ (e.g. $z = 1$). M alone decides it.

(d) φ_4 (z bound by $\exists$): true – $\exists z\,(z = 0)$ holds (witness $z = 0$). M alone decides it.

Pop Quiz Contd.

Note first: $Q(g(x,y), g(y,y), \cdot)$ collapses to "$(\cdot = 0)$", independent of x, y.

(a) φ_1 (constant c, c^M=0): true – M alone decides it, since c^M is already fixed at 0.

(b) φ_2 (free z): cannot be determined from M alone!
z is free, so we also need an environment α

- e.g. $\alpha(z)$:=0 gives $M, \alpha \models \varphi_2$, but $\alpha'(z)$:=1 gives $M, \alpha' \not\models \varphi_2$. Same M, two different answers.

(c) φ_3 (z bound by $\forall$): false – $\forall z\,(z = 0)$ fails in $\mathbb{N}$ (e.g. $z = 1$). M alone decides it.

(d) φ_4 (z bound by $\exists$): true – $\exists z\,(z = 0)$ holds (witness $z = 0$). M alone decides it.

Moral: $\varphi_1, \varphi_3, \varphi_4$ are all sentences – M alone fixes their truth value, no environment needed.
Only φ_2 has a free variable, so it genuinely needs an environment too.

Topic 17.1

Normal Forms for FOL

Why Normal Forms, Again?

- Recall (Module 1): propositional resolution and DPLL both needed the formula in CNF first.

Why Normal Forms, Again?

- Recall (Module 1): propositional resolution and DPLL both needed the formula in CNF first.
- FOL has the same story – but quantifiers complicate things: $\forall, \exists$ can be nested, mixed, and their variables reused.

Why Normal Forms, Again?

- Recall (Module 1): propositional resolution and DPLL both needed the formula in CNF first.
- FOL has the same story – but quantifiers complicate things: $\forall, \exists$ can be nested, mixed, and their variables reused.
- Goal today: strip away the quantifiers until what's left is propositional-shaped again –
  - Prenex form: push all quantifiers out front;
  - Skolemization: eliminate the $\exists$'s;
  - Clausal form: what remains is a set of clauses, just like propositional CNF.

Why Normal Forms, Again?

- Recall (Module 1): propositional resolution and DPLL both needed the formula in CNF first.
- FOL has the same story – but quantifiers complicate things: $\forall, \exists$ can be nested, mixed, and their variables reused.
- Goal today: strip away the quantifiers until what's left is propositional-shaped again –
  - Prenex form: push all quantifiers out front;
  - Skolemization: eliminate the $\exists$'s;
  - Clausal form: what remains is a set of clauses, just like propositional CNF.
- This is exactly the machinery we need for:
  1. FOL resolution
  2. DPLL extensions to handle FOL (SMT solvers)

Prenex Normal Form

Example: $\forall w \forall y \left(R(f(w), y) \wedge (\neg R(w, c) \vee \neg P(w))\right)$ – all quantifiers pulled out front, matrix left quantifier-free.

Prenex Normal Form

Example: $\forall w \forall y \left(R(f(w), y) \wedge (\neg R(w, c) \vee \neg P(w))\right)$ – all quantifiers pulled out front, matrix left quantifier-free.

Prenex form

A sentence is in prenex form if it looks like $Q_1 x_1 \ldots Q_n x_n \varphi$, where each $Q_i \in \{\forall, \exists\}$ and the quantifier-free φ is called the matrix.

Prenex Normal Form

Example: $\forall w \forall y \left(R(f(w), y) \wedge (\neg R(w, c) \vee \neg P(w))\right)$ – all quantifiers pulled out front, matrix left quantifier-free.

Prenex form

A sentence is in prenex form if it looks like $Q_1 x_1 \ldots Q_n x_n \varphi$, where each $Q_i \in \{\forall, \exists\}$ and the quantifier-free φ is called the matrix.

We do this in two steps

1. Rectify – rename apart, so no quantifier clashes.
2. Pull quantifiers to the front (via NNF)

Rectified Formulas

$\neg\big(\exists x \forall y\, R(x,y) \rightarrow \forall y \exists x\, (R(x,y) \wedge P(x))\big)$: x and y are each reused by two *different* quantifiers!

Rectified Formulas

$\neg\big(\exists x \forall y\, R(x,y) \rightarrow \forall y \exists x\, (R(x,y) \wedge P(x))\big)$: x and y are each reused by two *different* quantifiers!

Fully bracketed, to make scopes explicit: $\neg\Big(\big(\exists x\, (\forall y\, R(x,y))\big) \rightarrow \big(\forall y\, (\exists x\, (R(x,y) \wedge P(x)))\big)\Big)$ – the two scopes never overlap, they just share names.

Rectified Formulas

$\neg\big(\exists x \forall y\, R(x,y) \rightarrow \forall y \exists x\, (R(x,y) \wedge P(x))\big)$: x and y are each reused by two *different* quantifiers!

Fully bracketed, to make scopes explicit: $\neg\Big(\big(\exists x\, (\forall y\, R(x,y))\big) \rightarrow \big(\forall y\, (\exists x\, (R(x,y) \wedge P(x)))\big)\Big)$ – the two scopes never overlap, they just share names.

Rectified

A formula is rectified ("renamed apart") if no two quantifiers use the same variable, and no variable occurs both bound and free.

Rectified Formulas

$\neg\big(\exists x \forall y\, R(x,y) \rightarrow \forall y \exists x\, (R(x,y) \wedge P(x))\big)$: x and y are each reused by two *different* quantifiers!

Fully bracketed, to make scopes explicit: $\neg\Big(\big(\exists x\, (\forall y\, R(x,y))\big) \rightarrow \big(\forall y\, (\exists x\, (R(x,y) \wedge P(x)))\big)\Big)$ – the two scopes never overlap, they just share names.

Rectified

A formula is rectified ("renamed apart") if no two quantifiers use the same variable, and no variable occurs both bound and free.

Renamed apart: $\neg\big(\exists x \forall y\, R(x,y) \rightarrow \forall z \exists w\, (R(w,z) \wedge P(w))\big)$

Rectified Formulas

$\neg\big(\exists x \forall y\, R(x,y) \rightarrow \forall y \exists x\, (R(x,y) \wedge P(x))\big)$: x and y are each reused by two *different* quantifiers!

Fully bracketed, to make scopes explicit: $\neg\Big(\big(\exists x\, (\forall y\, R(x,y))\big) \rightarrow \big(\forall y\, (\exists x\, (R(x,y) \wedge P(x)))\big)\Big)$ – the two scopes never overlap, they just share names.

Rectified

A formula is rectified ("renamed apart") if no two quantifiers use the same variable, and no variable occurs both bound and free.

Renamed apart: $\neg\big(\exists x \forall y\, R(x,y) \rightarrow \forall z \exists w\, (R(w,z) \wedge P(w))\big)$

- Why OK? $\forall x.F(x) \equiv \forall y.F(y)$ whenever y is fresh (similarly for $\exists$) – renaming bound variables never changes meaning.

Rectified Formulas

$\neg\big(\exists x \forall y\, R(x,y) \to \forall y \exists x\, (R(x,y) \wedge P(x))\big)$: x and y are each reused by two *different* quantifiers!

Fully bracketed, to make scopes explicit: $\neg\Big(\big(\exists x\, (\forall y\, R(x,y))\big) \to \big(\forall y\, (\exists x\, (R(x,y) \wedge P(x)))\big)\Big)$ – the two scopes never overlap, they just share names.

Rectified

A formula is rectified ("renamed apart") if no two quantifiers use the same variable, and no variable occurs both bound and free.

Renamed apart: $\neg\big(\exists x \forall y\, R(x,y) \to \forall z \exists w\, (R(w,z) \wedge P(w))\big)$

- Why OK? $\forall x.F(x) \equiv \forall y.F(y)$ whenever y is fresh (similarly for $\exists$) – renaming bound variables never changes meaning.
- Why do it? Without it, pulling a quantifier to the front can capture a variable wrongly!
  - E.g. $P(x) \wedge \forall x\, Q(x)$ – here x is *free* in $P(x)$.

Rectified Formulas

$\neg\big(\exists x \forall y\, R(x,y) \to \forall y \exists x\, (R(x,y) \wedge P(x))\big)$: x and y are each reused by two *different* quantifiers!

Fully bracketed, to make scopes explicit: $\neg\Big(\big(\exists x\, (\forall y\, R(x,y))\big) \to \big(\forall y\, (\exists x\, (R(x,y) \wedge P(x)))\big)\Big)$ – the two scopes never overlap, they just share names.

Rectified

A formula is rectified ("renamed apart") if no two quantifiers use the same variable, and no variable occurs both bound and free.

Renamed apart: $\neg\big(\exists x \forall y\, R(x,y) \to \forall z \exists w\, (R(w,z) \wedge P(w))\big)$

- Why OK? $\forall x.F(x) \equiv \forall y.F(y)$ whenever y is fresh (similarly for $\exists$) – renaming bound variables never changes meaning.
- Why do it? Without it, pulling a quantifier to the front can capture a variable wrongly!
  - E.g. $P(x) \wedge \forall x\, Q(x)$ – here x is *free* in $P(x)$.
  - Naively pulling $\forall x$ to the front: $\forall x\, (P(x) \wedge Q(x))$ – now x in $P(x)$ is captured, turning "some fixed x" into "for all x"!

Rectified Formulas

$\neg\big(\exists x \forall y\, R(x,y) \to \forall y \exists x\, (R(x,y) \wedge P(x))\big)$: x and y are each reused by two *different* quantifiers!
Fully bracketed, to make scopes explicit: $\neg\Big(\big(\exists x\, (\forall y\, R(x,y))\big) \to \big(\forall y\, (\exists x\, (R(x,y) \wedge P(x)))\big)\Big)$ – the two scopes never overlap, they just share names.

Rectified

A formula is rectified ("renamed apart") if no two quantifiers use the same variable, and no variable occurs both bound and free.

Renamed apart: $\neg\big(\exists x \forall y\, R(x,y) \to \forall z \exists w\, (R(w,z) \wedge P(w))\big)$

- Why OK? $\forall x.F(x) \equiv \forall y.F(y)$ whenever y is fresh (similarly for $\exists$) – renaming bound variables never changes meaning.
- Why do it? Without it, pulling a quantifier to the front can capture a variable wrongly!
  - E.g. $P(x) \wedge \forall x\, Q(x)$ – here x is *free* in $P(x)$.
  - Naively pulling $\forall x$ to the front: $\forall x\, (P(x) \wedge Q(x))$ – now x in $P(x)$ is captured, turning "some fixed x" into "for all x"!
  - Correct: rename first ($\forall x\, Q(x) \equiv \forall x'\, Q(x')$), *then* pull: $\forall x'\, (P(x) \wedge Q(x'))$

Rectified Formulas

$\neg\big(\exists x \forall y\, R(x,y) \to \forall y \exists x\, (R(x,y) \wedge P(x))\big)$: x and y are each reused by two *different* quantifiers!

Fully bracketed, to make scopes explicit: $\neg\Big(\big(\exists x\, (\forall y\, R(x,y))\big) \to \big(\forall y\, (\exists x\, (R(x,y) \wedge P(x)))\big)\Big)$ – the two scopes never overlap, they just share names.

Rectified

A formula is rectified ("renamed apart") if no two quantifiers use the same variable, and no variable occurs both bound and free.

Renamed apart: $\neg\big(\exists x \forall y\, R(x,y) \to \forall z \exists w\, (R(w,z) \wedge P(w))\big)$

- Why OK? $\forall x.F(x) \equiv \forall y.F(y)$ whenever y is fresh (similarly for $\exists$) – renaming bound variables never changes meaning.
- Why do it? Without it, pulling a quantifier to the front can capture a variable wrongly!
  - E.g. $P(x) \wedge \forall x\, Q(x)$ – here x is *free* in $P(x)$.
  - Naively pulling $\forall x$ to the front: $\forall x\, (P(x) \wedge Q(x))$ – now x in $P(x)$ is captured, turning "some fixed x" into "for all x"!
  - Correct: rename first ($\forall x\, Q(x) \equiv \forall x'\, Q(x')$), *then* pull: $\forall x'\, (P(x) \wedge Q(x')) \equiv P(x) \wedge \forall x\, Q(x)$

Renaming Lemma

Renaming Lemma

If y does not occur in $\varphi(x)$, then $\forall x\, \varphi(x) \equiv \forall y\, \varphi(y)$ (similarly for $\exists$).

Renaming Lemma

Renaming Lemma

If y does not occur in $\varphi(x)$, then $\forall x\, \varphi(x) \equiv \forall y\, \varphi(y)$ (similarly for $\exists$).

Exercise 17.1 Prove it using ND!

Renaming Lemma

Renaming Lemma

If y does not occur in $\varphi(x)$, then $\forall x\, \varphi(x) \equiv \forall y\, \varphi(y)$ (similarly for $\exists$).

Exercise 17.1 Prove it using ND! (Recall $\forall e, \forall i$ from Lecture 14.)

Renaming Lemma

Renaming Lemma

If y does not occur in $\varphi(x)$, then $\forall x\, \varphi(x) \equiv \forall y\, \varphi(y)$ (similarly for $\exists$).

Exercise 17.1 Prove it using ND! (Recall $\forall e, \forall i$ from Lecture 14.) One direction:

1. $\forall x\, \varphi(x)$ premise
2.

Renaming Lemma

Renaming Lemma

If y does not occur in $\varphi(x)$, then $\forall x\, \varphi(x) \equiv \forall y\, \varphi(y)$ (similarly for $\exists$).

Exercise 17.1 Prove it using ND! (Recall $\forall e, \forall i$ from Lecture 14.) One direction:

1.	$\forall x\, \varphi(x)$	premise
2.	$y_0\ \varphi(y_0)$	$\forall x\, e$, 1
3.		

Renaming Lemma

Renaming Lemma

If y does not occur in $\varphi(x)$, then $\forall x\, \varphi(x) \equiv \forall y\, \varphi(y)$ (similarly for $\exists$).

Exercise 17.1 Prove it using ND! (Recall $\forall e, \forall i$ from Lecture 14.) One direction:

1.	$\forall x\, \varphi(x)$	premise
2.	$y_0\ \varphi(y_0)$	$\forall x\, e$, 1
3.	$\forall y\, \varphi(y)$	$\forall y\, i$, 2

(box scoped by fresh y_0 – the side conditions for both rules hold precisely because y_0 is fresh. The converse runs symmetrically.)

Renaming Lemma

Renaming Lemma

If y does not occur in $\varphi(x)$, then $\forall x\, \varphi(x) \equiv \forall y\, \varphi(y)$ (similarly for $\exists$).

Exercise 17.1 Prove it using ND! (Recall $\forall e, \forall i$ from Lecture 14.) One direction:

1.	$\forall x\, \varphi(x)$	premise
2.	$y_0\ \varphi(y_0)$	$\forall x\, e$, 1
3.	$\forall y\, \varphi(y)$	$\forall y\, i$, 2

(box scoped by fresh y_0 – the side conditions for both rules hold precisely because y_0 is fresh. The converse runs symmetrically.)

- **Exercise 17.2** adapt this to prove the $\exists$ case.

Substitution Needs Renaming Too

Recall (PS6, Q2d): $\varphi := \exists x\,(P(y,z) \wedge \forall y\,(\neg Q(y,x) \vee P(y,z)))$.

Substitution Needs Renaming Too

Recall (PS6, Q2d): $\varphi := \exists x\,(P(y,z) \wedge \forall y\,(\neg Q(y,x) \vee P(y,z)))$.

- Substituting t for a variable is only safe when t is free for that variable in φ

Substitution Needs Renaming Too

Recall (PS6, Q2d): $\varphi := \exists x\,(P(y,z) \wedge \forall y\,(\neg Q(y,x) \vee P(y,z)))$.

- Substituting t for a variable is only safe when t is free for that variable in φ
- Fix, when it's not: rename the offending bound variable first (Renaming Lemma!), *then* substitute.

Substitution Needs Renaming Too

Recall (PS6, Q2d): $\varphi := \exists x\,(P(y,z) \wedge \forall y\,(\neg Q(y,x) \vee P(y,z)))$.

- Substituting t for a variable is only safe when t is free for that variable in φ
- Fix, when it's not: rename the offending bound variable first (Renaming Lemma!), *then* substitute.

$\varphi[f(x)/y]$: free y sits inside $\exists x$'s scope, and $f(x)$ contains x

- Naive: $\exists x\,(P(f(x),z) \wedge \forall y(\neg Q(y,x) \vee P(y,z)))$ – the x in $f(x)$ gets captured by $\exists x$!

Substitution Needs Renaming Too

Recall (PS6, Q2d): $\varphi := \exists x\,(P(y,z) \wedge \forall y\,(\neg Q(y,x) \vee P(y,z)))$.

- Substituting t for a variable is only safe when t is free for that variable in φ
- Fix, when it's not: rename the offending bound variable first (Renaming Lemma!), *then* substitute.

$\varphi[f(x)/y]$: free y sits inside $\exists x$'s scope, and $f(x)$ contains x

- Naive: $\exists x\,(P(f(x),z) \wedge \forall y(\neg Q(y,x) \vee P(y,z)))$ – the x in $f(x)$ gets captured by $\exists x$!
- Rename first: $\exists x(\ldots) \equiv \exists x'\,(P(y,z) \wedge \forall y(\neg Q(y,x') \vee P(y,z)))$

Substitution Needs Renaming Too

Recall (PS6, Q2d): $\varphi := \exists x\,(P(y,z) \wedge \forall y\,(\neg Q(y,x) \vee P(y,z)))$.

- Substituting t for a variable is only safe when t is free for that variable in φ
- Fix, when it's not: rename the offending bound variable first (Renaming Lemma!), *then* substitute.

$\varphi[f(x)/y]$: free y sits inside $\exists x$'s scope, and $f(x)$ contains x

- Naive: $\exists x\,(P(f(x),z) \wedge \forall y(\neg Q(y,x) \vee P(y,z)))$ – the x in $f(x)$ gets captured by $\exists x$!
- Rename first: $\exists x(\ldots) \equiv \exists x'\,(P(y,z) \wedge \forall y(\neg Q(y,x') \vee P(y,z)))$
- Now substitute safely: $\exists x'\,(P(f(x),z) \wedge \forall y(\neg Q(y,x') \vee P(y,z)))$ – x in $f(x)$ stays free.

Substitution Needs Renaming Too

Recall (PS6, Q2d): $\varphi := \exists x\,(P(y,z) \wedge \forall y\,(\neg Q(y,x) \vee P(y,z)))$.

- Substituting t for a variable is only safe when t is free for that variable in φ
- Fix, when it's not: rename the offending bound variable first (Renaming Lemma!), *then* substitute.

$\varphi[f(x)/y]$: free y sits inside $\exists x$'s scope, and $f(x)$ contains x

- Naive: $\exists x\,(P(f(x),z) \wedge \forall y(\neg Q(y,x) \vee P(y,z)))$ – the x in $f(x)$ gets captured by $\exists x$!
- Rename first: $\exists x(\ldots) \equiv \exists x'\,(P(y,z) \wedge \forall y(\neg Q(y,x') \vee P(y,z)))$
- Now substitute safely: $\exists x'\,(P(f(x),z) \wedge \forall y(\neg Q(y,x') \vee P(y,z)))$ – x in $f(x)$ stays free.

$\varphi[g(y,z)/z]$: 2nd free z sits inside $\forall y$'s scope, and $g(y,z)$ contains y

- Naive: $\exists x\,(P(y,g(y,z)) \wedge \forall y(\neg Q(y,x) \vee P(y,g(y,z))))$ – the *2nd* $g(y,z)$'s y gets captured by that very $\forall y$!

Substitution Needs Renaming Too

Recall (PS6, Q2d): $\varphi := \exists x\,(P(y,z) \wedge \forall y\,(\neg Q(y,x) \vee P(y,z)))$.

- Substituting t for a variable is only safe when t is free for that variable in φ
- Fix, when it's not: rename the offending bound variable first (Renaming Lemma!), *then* substitute.

$\varphi[f(x)/y]$: free y sits inside $\exists x$'s scope, and $f(x)$ contains x

- Naive: $\exists x\,(P(f(x),z) \wedge \forall y(\neg Q(y,x) \vee P(y,z)))$ – the x in $f(x)$ gets captured by $\exists x$!
- Rename first: $\exists x(\ldots) \equiv \exists x'\,(P(y,z) \wedge \forall y(\neg Q(y,x') \vee P(y,z)))$
- Now substitute safely: $\exists x'\,(P(f(x),z) \wedge \forall y(\neg Q(y,x') \vee P(y,z)))$ – x in $f(x)$ stays free.

$\varphi[g(y,z)/z]$: 2nd free z sits inside $\forall y$'s scope, and $g(y,z)$ contains y

- Naive: $\exists x\,(P(y,g(y,z)) \wedge \forall y(\neg Q(y,x) \vee P(y,g(y,z))))$ – the *2nd* $g(y,z)$'s y gets captured by that very $\forall y$!
- Rename first: $\forall y(\ldots) \equiv \forall y'\,(\neg Q(y',x) \vee P(y',z))$

Substitution Needs Renaming Too

Recall (PS6, Q2d): $\varphi := \exists x\,(P(y,z) \wedge \forall y\,(\neg Q(y,x) \vee P(y,z)))$.

- Substituting t for a variable is only safe when t is free for that variable in φ
- Fix, when it's not: rename the offending bound variable first (Renaming Lemma!), *then* substitute.

$\varphi[f(x)/y]$: free y sits inside $\exists x$'s scope, and $f(x)$ contains x

- Naive: $\exists x\,(P(f(x),z) \wedge \forall y(\neg Q(y,x) \vee P(y,z)))$ – the x in $f(x)$ gets captured by $\exists x$!
- Rename first: $\exists x(\ldots) \equiv \exists x'\,(P(y,z) \wedge \forall y(\neg Q(y,x') \vee P(y,z)))$
- Now substitute safely: $\exists x'\,(P(f(x),z) \wedge \forall y(\neg Q(y,x') \vee P(y,z)))$ – x in $f(x)$ stays free.

$\varphi[g(y,z)/z]$: 2nd free z sits inside $\forall y$'s scope, and $g(y,z)$ contains y

- Naive: $\exists x\,(P(y,g(y,z)) \wedge \forall y(\neg Q(y,x) \vee P(y,g(y,z))))$ – the *2nd* $g(y,z)$'s y gets captured by that very $\forall y$!
- Rename first: $\forall y(\ldots) \equiv \forall y'\,(\neg Q(y',x) \vee P(y',z))$
- Now substitute safely: $\exists x\,(P(y,g(y,z)) \wedge \forall y'(\neg Q(y',x) \vee P(y',g(y,z))))$.

More Equivalences

Recall (Lecture 13): $\varphi \equiv \psi$ iff $\{\varphi\} \models \psi$ and $\{\psi\} \models \varphi$.

More Equivalences

Recall (Lecture 13): $\varphi \equiv \psi$ iff $\{\varphi\} \models \psi$ and $\{\psi\} \models \varphi$.

- Quantifier duals: $\neg\forall x\, \varphi \equiv \exists x\, \neg\varphi$, and $\neg\exists x\, \varphi \equiv \forall x\, \neg\varphi$
- Distributivity (*same* x on both sides): $\forall x\, (\varphi \wedge \psi) \equiv (\forall x\, \varphi) \wedge (\forall x\, \psi)$, and $\exists x\, (\varphi \vee \psi) \equiv (\exists x\, \varphi) \vee (\exists x\, \psi)$

More Equivalences

Recall (Lecture 13): $\varphi \equiv \psi$ iff $\{\varphi\} \models \psi$ and $\{\psi\} \models \varphi$.

- Quantifier duals: $\neg\forall x\,\varphi \equiv \exists x\,\neg\varphi$, and $\neg\exists x\,\varphi \equiv \forall x\,\neg\varphi$
- Distributivity (*same* x on both sides): $\forall x\,(\varphi \wedge \psi) \equiv (\forall x\,\varphi) \wedge (\forall x\,\psi)$, and $\exists x\,(\varphi \vee \psi) \equiv (\exists x\,\varphi) \vee (\exists x\,\psi)$

Exercise 17.3 Prove the following too – *(i) directly* (semantically), *and (ii) using ND*:

1. Pulling out of $\wedge$: $(Qx\,\varphi) \wedge \psi \equiv Qx\,(\varphi \wedge \psi)$, for $Q \in \{\forall, \exists\}$, provided $x \notin \mathit{free}(\psi)$
2. Pulling out of $\vee$: $(Qx\,\varphi) \vee \psi \equiv Qx\,(\varphi \vee \psi)$, for $Q \in \{\forall, \exists\}$, provided $x \notin \mathit{free}(\psi)$

More Equivalences

Recall (Lecture 13): $\varphi \equiv \psi$ iff $\{\varphi\} \models \psi$ and $\{\psi\} \models \varphi$.

- Quantifier duals: $\neg\forall x\,\varphi \equiv \exists x\,\neg\varphi$, and $\neg\exists x\,\varphi \equiv \forall x\,\neg\varphi$
- Distributivity (*same* x on both sides): $\forall x\,(\varphi \wedge \psi) \equiv (\forall x\,\varphi) \wedge (\forall x\,\psi)$, and $\exists x\,(\varphi \vee \psi) \equiv (\exists x\,\varphi) \vee (\exists x\,\psi)$

Exercise 17.3 Prove the following too – *(i) directly* (semantically), *and (ii) using ND*:

1. Pulling out of $\wedge$: $(Qx\,\varphi) \wedge \psi \equiv Qx\,(\varphi \wedge \psi)$, for $Q \in \{\forall, \exists\}$, provided $x \notin \mathit{free}(\psi)$
2. Pulling out of $\vee$: $(Qx\,\varphi) \vee \psi \equiv Qx\,(\varphi \vee \psi)$, for $Q \in \{\forall, \exists\}$, provided $x \notin \mathit{free}(\psi)$

Careful: in practice we only ever apply this to already-rectified formulas, where x doesn't occur in ψ *at all*.

Rectified Prenex Form (RPF)

A sentence is in Rectified Prenex Form if it is in prenex form ($Q_1 x_1 \dots Q_n x_n \varphi$) and additionally rectified: all the x_i are distinct, and none occurs free in φ.

Rectified Prenex Form (RPF)

A sentence is in Rectified Prenex Form if it is in prenex form ($Q_1 x_1 \ldots Q_n x_n \varphi$) and additionally rectified: all the x_i are distinct, and none occurs free in φ.

- To convert to RPF: push negations to the atoms (NNF), then pull all quantifiers to the front.
- Note: NNF is *not* part of RPF definition – it's just a consequence of the method to pull quantifiers out safely.

Rectified Prenex Form (RPF)

A sentence is in Rectified Prenex Form if it is in prenex form ($Q_1 x_1 \dots Q_n x_n \varphi$) and additionally rectified: all the x_i are distinct, and none occurs free in φ.

- To convert to RPF: push negations to the atoms (NNF), then pull all quantifiers to the front.
- Note: NNF is *not* part of RPF definition – it's just a consequence of the method to pull quantifiers out safely.

Exercise 17.4 Convert to RPF

$\neg\big(\exists x \forall y\, R(x,y) \to \forall z \exists w\, (R(w,z) \wedge P(w))\big)$

Rectified Prenex Form (RPF)

A sentence is in Rectified Prenex Form if it is in prenex form $(Q_1 x_1 \dots Q_n x_n \varphi)$ and additionally rectified: all the x_i are distinct, and none occurs free in φ.

- To convert to RPF: push negations to the atoms (NNF), then pull all quantifiers to the front.
- Note: NNF is *not* part of RPF definition – it's just a consequence of the method to pull quantifiers out safely.

Exercise 17.4 Convert to RPF

$$\neg\big(\exists x \forall y\, R(x,y) \to \forall z \exists w\, (R(w,z) \wedge P(w))\big)$$

$$\equiv \quad \exists x \forall y\, R(x,y) \quad \wedge \quad \neg \forall z \exists w\, (R(w,z) \wedge P(w)) \qquad (\neg(A \to B) \equiv A \wedge \neg B)$$

Rectified Prenex Form (RPF)

A sentence is in Rectified Prenex Form if it is in prenex form ($Q_1 x_1 \ldots Q_n x_n \, \varphi$) and additionally rectified: all the x_i are distinct, and none occurs free in φ.

- To convert to RPF: push negations to the atoms (NNF), then pull all quantifiers to the front.
- Note: NNF is *not* part of RPF definition – it's just a consequence of the method to pull quantifiers out safely.

Exercise 17.4 Convert to RPF

$\neg\big(\exists x \forall y \, R(x,y) \to \forall z \exists w \, (R(w,z) \wedge P(w))\big)$

$\equiv \; \exists x \forall y \, R(x,y) \; \wedge \; \neg \forall z \exists w \, (R(w,z) \wedge P(w))$ $\quad (\neg(A \to B) \equiv A \wedge \neg B)$

$\equiv \; \exists x \forall y \, R(x,y) \; \wedge \; \exists z \forall w \, \neg(R(w,z) \wedge P(w))$ $\quad$ (push $\neg$ through $\forall, \exists$: $\neg \forall z \, \varphi \equiv \exists z \, \neg \varphi$, $\neg \exists w \, \varphi \equiv \forall w \, \neg \varphi$)

Rectified Prenex Form (RPF)

A sentence is in Rectified Prenex Form if it is in prenex form ($Q_1 x_1 \dots Q_n x_n \, \varphi$) and additionally rectified: all the x_i are distinct, and none occurs free in φ.

- To convert to RPF: push negations to the atoms (NNF), then pull all quantifiers to the front.
- Note: NNF is *not* part of RPF definition – it's just a consequence of the method to pull quantifiers out safely.

Exercise 17.4 Convert to RPF

$\neg\big(\exists x \forall y \, R(x,y) \to \forall z \exists w \, (R(w,z) \wedge P(w))\big)$

$\equiv \; \exists x \forall y \, R(x,y) \; \wedge \; \neg \forall z \exists w \, (R(w,z) \wedge P(w))$ $\quad (\neg(A \to B) \equiv A \wedge \neg B)$

$\equiv \; \exists x \forall y \, R(x,y) \; \wedge \; \exists z \forall w \, \neg(R(w,z) \wedge P(w))$ $\quad$ (push $\neg$ through $\forall, \exists$: $\neg \forall z \, \varphi \equiv \exists z \, \neg \varphi$, $\neg \exists w \, \varphi \equiv \forall w \, \neg \varphi$)

$\equiv \; \exists x \forall y \, R(x,y) \; \wedge \; \exists z \forall w \, (\neg R(w,z) \vee \neg P(w))$ $\quad$ (De Morgan: $\neg(A \wedge B) \equiv \neg A \vee \neg B$)

Rectified Prenex Form (RPF)

A sentence is in Rectified Prenex Form if it is in prenex form ($Q_1 x_1 \dots Q_n x_n \, \varphi$) and additionally rectified: all the x_i are distinct, and none occurs free in φ.

- To convert to RPF: push negations to the atoms (NNF), then pull all quantifiers to the front.
- Note: NNF is *not* part of RPF definition – it's just a consequence of the method to pull quantifiers out safely.

Exercise 17.4 Convert to RPF

$\neg\big(\exists x \forall y \, R(x,y) \to \forall z \exists w \, (R(w,z) \wedge P(w))\big)$

$\equiv \; \exists x \forall y \, R(x,y) \; \wedge \; \neg \forall z \exists w \, (R(w,z) \wedge P(w))$ $\quad (\neg(A \to B) \equiv A \wedge \neg B)$

$\equiv \; \exists x \forall y \, R(x,y) \; \wedge \; \exists z \forall w \, \neg(R(w,z) \wedge P(w))$ $\quad$ (push $\neg$ through $\forall, \exists$: $\neg \forall z \, \varphi \equiv \exists z \, \neg \varphi$, $\neg \exists w \, \varphi \equiv \forall w \, \neg \varphi$)

$\equiv \; \exists x \forall y \, R(x,y) \; \wedge \; \exists z \forall w \, (\neg R(w,z) \vee \neg P(w))$ $\quad$ (De Morgan: $\neg(A \wedge B) \equiv \neg A \vee \neg B$)

$\equiv \; \exists z \, \forall w \, \exists x \, \forall y \, \big(R(x,y) \wedge (\neg R(w,z) \vee \neg P(w))\big)$ $\quad$ (pull quantifiers to the front one at a time – safe since rectified)

Skolemization

Example: Let $M(x)$: "x is a man", $W(x)$: "x is a woman". "For every man, there is a woman he likes": $\forall m \exists w \, (M(m) \rightarrow (W(w) \wedge \mathit{Likes}(m, w)))$.

Skolemization

Example: Let $M(x)$: "x is a man", $W(x)$: "x is a woman". "For every man, there is a woman he likes": $\forall m \exists w\,(M(m) \rightarrow (W(w) \wedge \mathit{Likes}(m, w)))$. Introduce a function f picking, for each man m, one such woman: $\forall m\,(M(m) \rightarrow (W(f(m)) \wedge \mathit{Likes}(m, f(m))))$.

Skolemization

Example: Let $M(x)$: "x is a man", $W(x)$: "x is a woman". "For every man, there is a woman he likes": $\forall m \exists w\,(M(m) \rightarrow (W(w) \wedge \mathit{Likes}(m, w)))$. Introduce a function f picking, for each man m, one such woman: $\forall m\,(M(m) \rightarrow (W(f(m)) \wedge \mathit{Likes}(m, f(m))))$.

The Skolemization Procedure

- In a prenex sentence, replace each $\exists x$ by a fresh function of all $\forall$-variables preceding it (just a fresh constant if none precede it), substituting it for x in the matrix, then drop the $\exists$.

Skolemization

Example: Let $M(x)$: "x is a man", $W(x)$: "x is a woman". "For every man, there is a woman he likes": $\forall m \exists w\,(M(m) \rightarrow (W(w) \wedge \mathit{Likes}(m, w)))$. Introduce a function f picking, for each man m, one such woman: $\forall m\,(M(m) \rightarrow (W(f(m)) \wedge \mathit{Likes}(m, f(m))))$.

The Skolemization Procedure

- In a prenex sentence, replace each $\exists x$ by a fresh function of all $\forall$-variables preceding it (just a fresh constant if none precede it), substituting it for x in the matrix, then drop the $\exists$.
- Such a function/constant is called a Skolem function/Skolem constant.

Skolemization

Example: Let $M(x)$: "x is a man", $W(x)$: "x is a woman". "For every man, there is a woman he likes": $\forall m \exists w\,(M(m) \rightarrow (W(w) \wedge \mathit{Likes}(m, w)))$. Introduce a function f picking, for each man m, one such woman: $\forall m\,(M(m) \rightarrow (W(f(m)) \wedge \mathit{Likes}(m, f(m))))$.

The Skolemization Procedure

- In a prenex sentence, replace each $\exists x$ by a fresh function of all $\forall$-variables preceding it (just a fresh constant if none precede it), substituting it for x in the matrix, then drop the $\exists$.
- Such a function/constant is called a Skolem function/Skolem constant.
- Work outermost $\exists$ first.

Skolemization

Example: Let $M(x)$: "x is a man", $W(x)$: "x is a woman". "For every man, there is a woman he likes": $\forall m \exists w\,(M(m) \rightarrow (W(w) \wedge \mathit{Likes}(m, w)))$. Introduce a function f picking, for each man m, one such woman: $\forall m\,(M(m) \rightarrow (W(f(m)) \wedge \mathit{Likes}(m, f(m))))$.

The Skolemization Procedure

- In a prenex sentence, replace each $\exists x$ by a fresh function of all $\forall$-variables preceding it (just a fresh constant if none precede it), substituting it for x in the matrix, then drop the $\exists$.
- Such a function/constant is called a Skolem function/Skolem constant.
- Work outermost $\exists$ first.

Continuing Exercise 17.4's RPF: $\exists z \forall w \exists x \forall y\,(R(x, y) \wedge (\neg R(w, z) \vee \neg P(w)))$.

- No $\forall$ before $\exists z$: introduce constant c: $\forall w \exists x \forall y\,(R(x, y) \wedge (\neg R(w, c) \vee \neg P(w)))$

Skolemization

Example: Let $M(x)$: "x is a man", $W(x)$: "x is a woman". "For every man, there is a woman he likes": $\forall m \exists w\,(M(m) \rightarrow (W(w) \wedge \mathit{Likes}(m,w)))$. Introduce a function f picking, for each man m, one such woman: $\forall m\,(M(m) \rightarrow (W(f(m)) \wedge \mathit{Likes}(m, f(m))))$.

The Skolemization Procedure

- In a prenex sentence, replace each $\exists x$ by a fresh function of all $\forall$-variables preceding it (just a fresh constant if none precede it), substituting it for x in the matrix, then drop the $\exists$.
- Such a function/constant is called a Skolem function/Skolem constant.
- Work outermost $\exists$ first.

Continuing Exercise 17.4's RPF: $\exists z \forall w \exists x \forall y\,(R(x,y) \wedge (\neg R(w,z) \vee \neg P(w)))$.

- No $\forall$ before $\exists z$: introduce constant c: $\forall w \exists x \forall y\,(R(x,y) \wedge (\neg R(w,c) \vee \neg P(w)))$
- $\forall w$ precedes $\exists x$: introduce $f(w)$: $\forall w \forall y\,(R(f(w),y) \wedge (\neg R(w,c) \vee \neg P(w)))$

Skolemization

Example: Let $M(x)$: "x is a man", $W(x)$: "x is a woman". "For every man, there is a woman he likes": $\forall m \exists w\,(M(m) \rightarrow (W(w) \wedge \mathit{Likes}(m,w)))$. Introduce a function f picking, for each man m, one such woman: $\forall m\,(M(m) \rightarrow (W(f(m)) \wedge \mathit{Likes}(m, f(m))))$.

The Skolemization Procedure

- In a prenex sentence, replace each $\exists x$ by a fresh function of all $\forall$-variables preceding it (just a fresh constant if none precede it), substituting it for x in the matrix, then drop the $\exists$.
- Such a function/constant is called a Skolem function/Skolem constant.
- Work outermost $\exists$ first.

Continuing Exercise 17.4's RPF: $\exists z \forall w \exists x \forall y\,(R(x,y) \wedge (\neg R(w,z) \vee \neg P(w)))$.

- No $\forall$ before $\exists z$: introduce constant c: $\forall w \exists x \forall y\,(R(x,y) \wedge (\neg R(w,c) \vee \neg P(w)))$
- $\forall w$ precedes $\exists x$: introduce $f(w)$: $\forall w \forall y\,(R(f(w),y) \wedge (\neg R(w,c) \vee \neg P(w)))$

Skolem Normal Form

- Skolemization results in sentence with only universal quantifiers: Skolem Normal Form (SNF)
- SNF is equisatisfiable with the original formula, though generally *not* equivalent to it.

Clausal Form (for Resolution)

Example: continuing $\forall w \forall y \left(R(f(w), y) \wedge (\neg R(w, c) \vee \neg P(w))\right)$ from before.

Clausal Form (for Resolution)

Example: continuing $\forall w \forall y \left(R(f(w), y) \wedge (\neg R(w, c) \vee \neg P(w))\right)$ from before.

- Finally, make the matrix into CNF – distribute $\vee$ over $\wedge$ wherever needed!
  - Here $R(f(w), y) \wedge (\neg R(w, c) \vee \neg P(w))$ is *already* a conjunction of clauses – nothing to do!

Clausal Form (for Resolution)

Example: continuing $\forall w \forall y \left(R(f(w), y) \wedge (\neg R(w, c) \vee \neg P(w))\right)$ from before.

- Finally, make the matrix into CNF – distribute $\vee$ over $\wedge$ wherever needed!
  - Here $R(f(w), y) \wedge (\neg R(w, c) \vee \neg P(w))$ is *already* a conjunction of clauses – nothing to do!
- Since $\forall$ distributes over $\wedge$:

$$(\forall w \forall y\, R(f(w), y)) \;\wedge\; (\forall w \forall y\, (\neg R(w, c) \vee \neg P(w)))$$

Clausal Form (for Resolution)

Example: continuing $\forall w \forall y \left(R(f(w), y) \wedge (\neg R(w, c) \vee \neg P(w)) \right)$ from before.

- Finally, make the matrix into CNF – distribute $\vee$ over $\wedge$ wherever needed!
  - Here $R(f(w), y) \wedge (\neg R(w, c) \vee \neg P(w))$ is *already* a conjunction of clauses – nothing to do!
- Since $\forall$ distributes over $\wedge$:

$$(\forall w \forall y \, R(f(w), y)) \;\wedge\; (\forall w \forall y \, (\neg R(w, c) \vee \neg P(w)))$$

- Dropping the (now implicit) universal quantifiers, we get a set of clauses:

$$\{ \, R(f(w), y), \quad \neg R(w, c) \vee \neg P(w) \, \}$$

Clausal Form (for Resolution)

Example: continuing $\forall w \forall y \left(R(f(w), y) \wedge (\neg R(w, c) \vee \neg P(w))\right)$ from before.

- Finally, make the matrix into CNF – distribute $\vee$ over $\wedge$ wherever needed!
  - Here $R(f(w), y) \wedge (\neg R(w, c) \vee \neg P(w))$ is *already* a conjunction of clauses – nothing to do!
- Since $\forall$ distributes over $\wedge$:

$$(\forall w \forall y\, R(f(w), y)) \;\wedge\; (\forall w \forall y\, (\neg R(w, c) \vee \neg P(w)))$$

- Dropping the (now implicit) universal quantifiers, we get a set of clauses:

$$\{\, R(f(w), y),\quad \neg R(w, c) \vee \neg P(w) \,\}$$

Clausal form

A sentence is in clausal form if written as a set of clauses $C_1, \ldots, C_k$ with *all* variables implicitly universally quantified, and no quantifier symbols remain.

Clausal Form (for Resolution)

Example: continuing $\forall w \forall y \left(R(f(w), y) \wedge (\neg R(w, c) \vee \neg P(w))\right)$ from before.

- Finally, make the matrix into CNF – distribute $\vee$ over $\wedge$ wherever needed!
  - Here $R(f(w), y) \wedge (\neg R(w, c) \vee \neg P(w))$ is *already* a conjunction of clauses – nothing to do!
- Since $\forall$ distributes over $\wedge$:

$$(\forall w \forall y\, R(f(w), y)) \;\wedge\; (\forall w \forall y\, (\neg R(w, c) \vee \neg P(w)))$$

- Dropping the (now implicit) universal quantifiers, we get a set of clauses:

$$\{\, R(f(w), y), \quad \neg R(w, c) \vee \neg P(w) \,\}$$

Clausal form

A sentence is in clausal form if written as a set of clauses $C_1, \ldots, C_k$ with *all* variables implicitly universally quantified, and no quantifier symbols remain.

Careful: different clauses never share variables, even if they reuse a name – e.g. the w in $R(f(w), y)$ is not the same w as in $\neg R(w, c) \vee \neg P(w)$.

Clausal Form (for Resolution)

Example: continuing $\forall w \forall y \left(R(f(w), y) \wedge (\neg R(w, c) \vee \neg P(w))\right)$ from before.

- Finally, make the matrix into CNF – distribute $\vee$ over $\wedge$ wherever needed!
  - Here $R(f(w), y) \wedge (\neg R(w, c) \vee \neg P(w))$ is *already* a conjunction of clauses – nothing to do!
- Since $\forall$ distributes over $\wedge$:

$$(\forall w \forall y\, R(f(w), y)) \;\wedge\; (\forall w \forall y\, (\neg R(w, c) \vee \neg P(w)))$$

- Dropping the (now implicit) universal quantifiers, we get a set of clauses:

$$\{\, R(f(w), y),\quad \neg R(w, c) \vee \neg P(w) \,\}$$

Clausal form

A sentence is in clausal form if written as a set of clauses $C_1, \ldots, C_k$ with *all* variables implicitly universally quantified, and no quantifier symbols remain.

Careful: different clauses never share variables, even if they reuse a name – e.g. the w in $R(f(w), y)$ is not the same w as in $\neg R(w, c) \vee \neg P(w)$.

- This is exactly the input format for FOL resolution, even if we won't go into that!