First-Order Logic

Part two

Logic in Computer Science

1

Prenex Normal Form

- A formula containing no quantifiers at all, or
- A formula of the form

$$Q_1x_1\ Q_2x_2\ \ldots\ Q_nx_n\ P$$

where Q_i are either the universal or existential quantifier, x_i are variables and P is free of quantifiers.

e.g., $\exists x\ \forall y\ (p(x) \rightarrow q(y))$.

2

Conversion to Prenex Normal Form

1. Replace implications, biconditionals, etc., by and-or-negation. E.g., $(A \rightarrow B)$ by $(\neg A \vee B)$
2. Move $\neg$ "inwards" until there are no quantifiers in the scope of a negation, by deMorgan's laws.
3. Rename variables so each variable following a quantifier has a unique name.
4. Move quantifiers to the front of the sentence, without changing their order.

• Prenex normal forms are not unique

3

Example of Prenex NF

$\forall x\, ((C(x) \wedge \exists y\, (T(y) \wedge L(x, y))) \rightarrow \exists y\, (D(y) \wedge B(x, y)))$

$\forall x\, (\neg(C(x) \wedge \exists y\, (T(y) \wedge L(x, y))) \vee \exists z\, (D(z) \wedge B(x, z)))$

$\forall x\, (\neg \exists y\, (C(x) \wedge T(y) \wedge L(x, y)) \vee \exists z\, (D(z) \wedge B(x, z)))$

$\forall x \forall y\, (\neg(C(x) \wedge T(y) \wedge L(x, y)) \vee \exists z\, (D(z) \wedge B(x, z)))$

$\forall x \forall y \exists z\, (\neg(C(x) \wedge T(y) \wedge L(x, y)) \vee (D(z) \wedge B(x, z)))$

If you want to restore the implication:

$\forall x \forall y \exists z\, (C(x) \wedge T(y) \wedge L(x, y)) \rightarrow (D(z) \wedge B(x, z))$

Another prenex normal form is:

$\forall x \exists z \forall y\, (C(x) \wedge T(y) \wedge L(x, y)) \rightarrow (D(z) \wedge B(x, z))$

4

Skolemization: Removal of Quantifiers

1. Obtain a Prenex NF $B = Q_1x_1\ Q_2x_2 \dots Q_nx_n\ P$
2. For j := 1 to n do
3. If (Q_j is $\forall$) remove $Q_j\ x_j$ from B
4. If (Q_j is $\forall\exists$) remove $Q_j\ x_j$ and replace x_j by f(V), where V is the set of free variables in B

Example: $A = \forall x \exists z \forall y\ (C(x) \wedge T(y) \wedge L(x,y)) \rightarrow (D(z) \wedge B(x,z))$

B := A

1. $B := \exists z \forall y\ (C(x) \wedge T(y) \wedge L(x, y)) \rightarrow (D(z) \wedge B(x, z))$
2. $B := \forall y\ (C(x) \wedge T(y) \wedge L(x, y)) \rightarrow (D(f(x)) \wedge B(x, f(x)))$
3. $B := (C(x) \wedge T(y) \wedge L(x, y)) \rightarrow (D(f(x)) \wedge B(x, f(x)))$

- **Theorem**: $A \approx B$, i.e., A and B are equally satisfiable.

5

CNF: Conjunction Normal Forms

1. Obtain a PNF of A: $B = Q_1x_1\ Q_2x_2 \dots Q_nx_n\ P$
2. Remove quantifiers by Skolemization
3. Convert the formula into CNF as in PL

Example:

- $A = \forall x \exists z \forall y\ (C(x) \wedge T(y) \wedge L(x, y)) \rightarrow (D(z) \wedge B(x, z))$
- $B = (C(x) \wedge T(y) \wedge L(x, y)) \rightarrow (D(f(x)) \wedge B(x, f(x)))$
- C = { (-C(x) | -T(y) | -L(x, y) | D(f(x)),
  (-C(x) | -T(y) | -L(x, y) | B(x, f(x)) }

- **Theorem**: $A \approx C$, i.e., A and C are equally satisfiable.

6

CNF: No Need to go through PNF

1. Obtain a NNF of A: B = NNF(A)
2. Remove quantifiers by Skolemization
3. Convert the formula into CNF as in PL

Example:

- $A = \neg\forall x\exists z\forall y\ (C(x) \wedge T(y) \wedge L(x, y)) \rightarrow (D(z) \wedge B(x, z))$
- $B = \exists x\forall z\exists y\ (C(x) \wedge T(y) \wedge L(x, y)) \wedge (\neg D(z) \vee \neg B(x, z))$
- $C = (C(c) \wedge T(f(z)) \wedge L(c, f(z))) \wedge (\neg D(z) \vee \neg B(c, z))$
- **C is already a CNF.**

- **Theorem**: $A \approx C$, i.e., A and C are equally satisfiable.

7

Definition of ≈

- We write $A \approx B$ to denote that A is satisfiable iff B is satisfiable.
- $A \equiv B$ implies $A \approx B$, but the inverse is not true.

Example: Consider $A = \exists y\ p(x, y)$ and $B = p(x, f(y))$.

For the interpretation $\mathbb{I} = (Z, \{>\}, \{f\})$, where Z is the set of all integers and $f(x) = x+1$, A is true in $\mathbb{I}$, but B is false in $\mathbb{I}$. So it's not true that $A \equiv B$.

(only-if part) Suppose A is true in $\mathbb{I} = (D, \{p\}, \{\})$. We extend $\mathbb{I}$ to $\mathbb{I}'$ by introducing a function f: $D \rightarrow D$ such that $f(d_1) = d_2$ if $p(d_1, d_2)$ is true in $\mathbb{I}' = (D, \{p\}, \{f\})$.

(if-part) If $\mathbb{I}'' = (D, \{p\}, \{f\})$ is a model of B, then it is easy to see that A is also true in $\mathbb{I}''$.

So it is true that $A \approx B$.

8

Converting formulas to CNF

1. Obtain NNF (negation normal form) A
   a. Get rid of ↔ or ⊕
   b. Get rid of →
   c. Push ¬ downward
2. Remove quantifiers by Skolemization to get B
   a. Rename quantified variables
   b. Replace existentially quantified variables by Skolem constants/functions.
   c. Discard all universal quantifiers
3. Convert B into clause set C
   a. Convert B into CNF
   b. Convert CNF into clause set
   c. Standardize the variables in clauses

9

Converting formulas to CNF

1a. Eliminate all ↔ connectives

$(P \leftrightarrow Q) \Rightarrow ((P \rightarrow Q) \wedge (Q \rightarrow P))$

1b. Eliminate all → connectives

$(P \rightarrow Q) \Rightarrow (\neg P \vee Q)$

1c. Reduce the scope of each negation symbol to a single predicate

$\neg\neg P \Rightarrow P$

$\neg(P \vee Q) \Rightarrow \neg P \wedge \neg Q$

$\neg(P \wedge Q) \Rightarrow \neg P \vee \neg Q$

$\neg\forall x\ P \Rightarrow \exists x\ \neg P$

$\neg \exists x\ P \Rightarrow \forall x\ \neg P$

10

Converting formulas to clausal form

Skolem constants and functions

2a. Standardize variables: rename all variables so that each quantifier has its own unique variable name

2b. Replace existential quantified variables by introducing Skolem constants or functions

$\exists x\ P(x)$ is changing to $P(C)$

C is a Skolem constant (a brand-new constant symbol that is not used in any other sentence)

$\forall x\ \exists y\ P(x,y)$ is changing to $P(x, f(x))$

since $\exists$ is within scope of a universally quantified variable, use a **Skolem function f** to construct a new value that **depends on** the universally quantified variable.

f must be a brand-new function name not occurring anywhere

11

Converting formulas to clausal form

2c. Remove universal quantifiers by (1) moving them all to the left end; (2) making the scope of each the entire sentence; and (3) dropping the "prefix" part

Example: $\forall x\ P(x)$ is changing to $P(x)$

3a. Put into conjunctive normal form (conjunction of disjunctions) using distributive and associative laws

$(P \wedge Q) \vee R \Rightarrow (P \vee R) \wedge (Q \vee R)$

$(P \vee Q) \vee R \Rightarrow (P \vee Q \vee R)$

3b. Split conjuncts into separate clauses

3c. Standardize variables so each clause contains only variable names that do not occur in any other clause

12

An example

$\forall x\ (P(x) \rightarrow ((\forall y)(P(y) \rightarrow P(f(x,y))) \wedge \neg(\forall y)(Q(x,y) \rightarrow P(y))))$

1a. Eliminate $\leftrightarrow$

1b. Eliminate $\rightarrow$

$\forall x\ (\neg P(x) \vee (\forall y\ (\neg P(y) \vee P(f(x, y))) \wedge \neg\forall y\ (\neg Q(x, y) \vee P(y))))$

1c Reduce scope of negation

$\forall x\ (\neg P(x) \vee (\forall y\ (\neg P(y) \vee P(f(x, y))) \wedge \exists y\ (Q(x, y) \wedge \neg P(y))))$

2a. Standardize variables

$\forall x\ (\neg P(x) \vee (\forall y\ (\neg P(y) \vee P(f(x, y))) \wedge \exists z\ (Q(x, z) \wedge \neg P(z))))$

2b. Eliminate existential quantification

$\forall x\ (\neg P(x) \vee(\forall y\ (\neg P(y) \vee P(f(x, y))) \wedge(Q(x, g(x)) \wedge \neg P(g(x)))))$

2c. Drop universal quantification symbols

$(\neg P(x) \vee ((\neg P(y) \vee P(f(x, y))) \wedge (Q(x, g(x)) \wedge \neg P(g(x)))))$

13

An Example (continued)

3a. Convert to conjunction of disjunctions

$(\neg P(x) \mid \neg P(y) \mid P(f(x,y))) \wedge (\neg P(x) \mid Q(x,g(x))) \wedge$
$(\neg P(x) \mid \neg P(g(x)))$

3b. Create separate clauses

$(\neg P(x) \mid \neg P(y) \mid P(f(x,y)))$

$(\neg P(x) \mid Q(x, g(x)))$

$(\neg P(x) \mid \neg P(g(x)))$

3c. Standardize variables

$(\neg P(x) \mid \neg P(y) \mid P(f(x,y)))$

$(\neg P(z) \mid Q(z, g(z)))$

$(\neg P(w) \mid \neg P(g(w)))$

14

Question: Given a finite set *F* of function symbols, and an infinite set *C* of constants, how a ground term built on them is enumerated?

Answer: Using weighted strings.

- Suppose $C = \{a_1, a_2, \ldots, a_i, \ldots\}$, $w(a_i) = i$.
- For each symbol f in *F*, w(f) = 1.
- For any given ground term t built on *F* and *C*, w(t) = the sum of all weights of symbols in t.
- To enumerate t, we enumerate all terms of weight ≤ w(t).
- It's guaranteed that every ground term will be enumerated.

15

Herbrand Models

- First-order language L = (*P, F, X, Op*)
- The models of L is I = (*D, R, G*)
- The models of FOL can be very complicated since we have infinite many choices for choosing a domain, a relation for a predicate symbol, a function for a function symbol.
- We will show that, if a set of clauses has a model (i.e. it is satisfiable), it has a particular **canonical** (or, generic) model, which is called **Herbrand model**.

19

Herbrand Universe

- First-order language L = (*P, F, X, Op*)
- The models of L is I = (*D, R, G*)
- S: set of clauses of L
- Herbrand Universe of S: H_S = T(*F*), the set of ground terms built on *F*, assuming *F* contains some constant symbols (otherwise, we add a constant c into *F*).
- Example: *F* = { c/0, s/1 }
  - H_S = T(*F*) = { c, s(c), s(s(c)), …, s^i(c), … }
- The set is H_S not empty and is infinite if contains a non-constant function symbol.

20

Herbrand Base

- First-order language L = (*P, F, X, Op*)
- The models of L is I = (*D, R, G*)
- S: set of clauses of L
- Herbrand Universe of S: H_S
- Herbrand Base of S: B_S is the set of all ground atomic formulas.
- Example: *F*= {a, b, f}, *P* = { p }, and

  S = {(¬p(a, f(x))), (p(b, f(y)))}

H_S = { a, b, f(a), f(b), f(f(a)), f(f(b)), f(f(f(a))), … }

B_S = { p(a, a), p(a, b), p(b, a), p(b, b), p(a, f(a)), p(a, f(b)), p(b, f(a)), p(b, f(b)), p(f(a), a), p(f(b), a), … }

21

Herbrand Models

- First-order language L = (*P, F, X, Op*)
- The models of L is I = (*D, R, G*)
- S: set of clauses of L
- Herbrand Universe of S: H_S
- Herbrand Base of S: B_S is the set of all ground atomic formulas.
- Herbrand Model of S: M_S is merely a subset of B_S, with the assumption that *D* = H_S = T(*F*), *G* = *F*, and *R* = *P* defined by M_S.

22

Herbrand Models

- Herbrand Model of S: M_S is merely a subset of B_S, with the assumption that *D* = H_S = T(*F*), *G* = *F*, and *R* is defined by M_S.
- The domain of the Herbrand model is the Herbrand universe H_S.
- *G* = *F*: For any f in *F*, $f(t_1, t_2, \ldots, t_k)$ is the result of applying f to $(t_1, t_2, \ldots, t_k)$ in $T^k(F)$.
- *R* is defined by M_S: For any p in *P*, $p(t_1, t_2, \ldots, t_k)$ is true iff $p(t_1, t_2, \ldots, t_k)$ is in M_S.

23

Herbrand Models

- Herbrand Model of S: M_S is merely a subset of B_S, with the assumption that $D = H_S = T(F)$, $G = F$, and R is defined by M_S.
- Example: $F = \{a, b, f\}$, $P = \{ p \}$, and

  $S = \{(\neg p(a, f(x))), (p(b, f(y)))\}$

$H_S = \{ a, b, f(a), f(b), f(f(a)), f(f(b)), f(f(f(a))), \ldots \}$

$B_S = \{ p(a, a), p(a, b), p(b, a), p(b, b), p(a, f(a)), p(a, f(b)), p(b, f(a)), p(b, f(b)), p(f(a), a), p(f(b), a), \ldots \}$

- $M_{S1} = \{ p(b, f(t)) \mid t \in H_S \}$ – the minimal H-model
- $M_{S2} = B_S - \{ p(a, f(t)) \mid t \in H_S \}$ – the maximal model.
- Any set M, where $M_{S1} \subseteq M \subseteq M_{S2}$, is a H-model.

24

Herbrand Theorem

- **Theorem:** Let S be a set of clauses. S has a model if and only if it has a Herbrand model.
  - The proof is given in the book.
- **Herbrand's Theorem**: A set of clauses S is unsatisfiable if and only if a **finite** set of **ground instances of clauses** from S is unsatisfiable.
  - The proof is omitted in the book.
- **Example**: $C = \{ (\neg p(x) \mid q(x)), (p(y)), (\neg q(z)) \}$
- One set of ground instances for this set of clauses is:

  $S = \{ (\neg p(a) \mid q(a)), (p(a)), (\neg q(a)) \}$

Unit resolution can show S is unsatisfiable.

25

Ground Resolution Rule

- Ground Resolution: $(p(t) \mid A), (-p(t) \mid C) \vdash (A \mid C)$
- $(p(t) \mid A), (-\,p(t) \mid C)$ are the *parents* of $(A \mid C)$;
- $(A \mid C)$ is their *resolvent* on the clashing literal $p(t)$
- Notation: $\mathrm{Res}((p(t) \mid A), (-p(t) \mid C)) = (A \mid C)$

- **Example**: S = { (¬p(a) | q(a)), (p(a)), (¬q(a)) }
- Res((¬p(a) | q(a)), (p(a))) = (q(a) | q(a))
- Res((q(a)), (¬q(a))) = ( ), the empty clause.

- **Theorem:** $(p(t) \mid A), (-p(t) \mid C) \models (A \mid C)$

26

Different Forms of Resolution

- Binary Resolution

$$\frac{(C_1 \mid A) \qquad (C_2 \mid -A)}{(C_1 \mid C_2)}$$

- Unit Resolution (when C_1 or C_2 is empty)

$$\frac{(\,p\,A\,) \qquad C_2 \mid -A)}{C_2} \qquad\qquad \frac{(C_1 \mid A) \qquad (-A)}{C_1}$$

- Clashing (when both C_1 and C_2 are empty)

$$\frac{(A) \qquad (-A)}{(\,)}$$

( ) is the empty clause, also denoted by □ or 0.

- As a refutation prover, ( ) is the goal.

27

Semi-Decision for FOL

- To decide if a FOL formula A is valid:
  1. Negate the formula: B = ¬ A.
  2. Transform B into a clausal form: C = CNF(B).
  3. Generate a finite set of ground instances of C: S = finiteInstances(C)
  4. Check if S is unsatisfiable by resolution.
- Step 3 is highly problematic: there are infinitely many ground terms (if there is at least one function symbol) so it is be difficult to find a correct subset of ground instances.
- It is a semi-decision procedure because if the set of clauses is satisfiable, it may have an infinite model.
- The validity problem in FOL is undecidable.

28

Colonel West is a criminal

1. It is a crime for an American to sell weapons to a hostile country.
2. The country Nono has some missiles.
3. All of its missiles were sold to it by Colonel West.
4. Nono is an enemy of USA.
5. Colonel West is an American.

29

Modeling with Horn Clauses: at most one positive literal

$(\neg A_1 \mid \neg A_2 \mid \neg A_3 \mid \neg A_4 \mid B)$ as $A_1 \wedge A_2 \wedge A_3 \wedge A_4 \rightarrow B$

1. It is a crime for an American to sell weapons to a hostile country.

   American(x) $\wedge$ Weapons(y) $\wedge$ Hostile(z) $\wedge$ Sell(x,y,z) $\rightarrow$ Criminal (x)

2. The country Nono has some missiles.

   // $\exists$ x Owns(Nono, x) $\wedge$ Missile(x)

   Missile(M1) // M1 is a Skolem Constant

   Owns(Nono, M1)

30

Modeling with Horn Clauses: at most one positive literal

3. All of its missiles were sold to it by Colonel West.

   Missile(x) $\wedge$ Owns(Nono, x) $\rightarrow$ Sells(West,x, Nono).

4. Nono is an enemy of USA.

   Enemy(Nono,Amer ican).

5. Colonel West is an American.

   American(West).

// common sense

Missile(x) $\rightarrow$ Weapon(x)

Enemy(x, America) $\rightarrow$ Hostile(x)

31