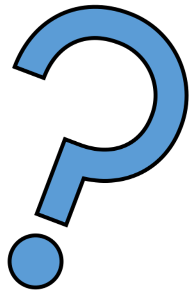
COMP9414 Tutorial

Week 9

News

- Assignment 2 is due tomorrow at midnight
    - Week 9, Friday, July 31^{st}, 11:59pm

- Ignore questions 2ii, 2iii and 2iv from tutorial 8
    - Wayne didn't want us to go over it
    - No idea why he included it



First-order Logic

- Express knowledge about objects, their properties and the relationships that they share

- Predicate symbols applied to various terms/variables
  - Combined with operators to provide meaning: $\forall, \exists, \rightarrow$

First-order Logic - Operators

$\forall x$	Universal	For all x
$\exists x$	Existential	At least one x
$\rightarrow$	Binary Connective	Leads to or is true as a result

Equality	$=, \neq, \leq, \geq$
Logical Connectives	$\wedge, \vee, \rightarrow, \leftrightarrow, \neg$

First-order Logic – Basic Examples

Term	Predicate	Operator

Someone likes James	$\exists x\ likes(x, James)$

The ∀ symbol generally involves a → symbol.	
All people are mortal	$\forall x\ (person(x) \rightarrow mortal(x))$

The ∃ symbol generally involves a ∧ symbol.	
All cats are immortal	$\exists x\ (cat(x) \land \neg mortal(x))$

First-order Logic – Equivalency Examples

Term	Predicate
Operator	

Every student took an exam	$\forall x\ (student(x) \rightarrow \exists y(exam(y)\ \wedge took(x, y)))$
Every student took an exam	$\exists y\ (exam(y) \wedge \forall x\ (student(x) \rightarrow took(x, y)))$

Brothers are siblings	$\forall x\ \forall y\ (brothers(x, y) \rightarrow sibling(x, y))$
Siblings	$\exists x\ \exists y\ (sibling(x, y) \leftrightarrow sibling(y, x))$

First-order Logic – Nested Quantifier Examples

Term	
Predicate	
Operator	

Everything likes everything	$\forall x \; \forall y \; likes(x, y)$
Something likes something	$\exists x \; \exists y \; likes(x, y)$
Everything likes something	$\forall x \; \exists y \; likes(x, y)$
Something liked by everything	$\exists x \; \forall y \; likes(x, y)$

Question 1i

$\forall x\ (bird(x) \rightarrow flies(x))$

$\forall x$	For all x
$bird(x)$	x is a bird
$flies(x)$	x can fly

For all x, x is a bird implies that x can fly
All birds can fly

Question 1iii

$\exists x \, \forall y \; (person(x) \wedge mother(x, y))$

$\exists x$	At least one x
$\forall y$	For all y

$person(x)$	x is a person
$mother(x, y)$	x is the mother of y

At least one x, for all y, x is a person and the mother of y
There is someone who is everyone's mother

Question 2i

All cats are mammals

Terms	N/A
Quantifiers	All
Predicates	cats, mammals
Operators	are

$\forall x\ (cat(x) \rightarrow mammal(x))$

Question 2iii

All computer scientists like some operating system

Terms	N/A
Quantifiers	All, some
Predicates	computer_scientist, operating_system, likes
Operators	N/A

$$\forall x\ (computer\_scientist(x) \rightarrow \exists y\ (operating\_system(y) \land likes(x, y)))$$

Background – Conjunctive Normal Form (CNF)

Conjunction of disjunctive literals	
$(A \vee B) \wedge (C \vee D) \wedge (E \vee F \vee G) \wedge H$	$\{A \vee B,\ C \vee D,\ E \vee F \vee G,\ H\}$
Grammar = {predicates, literals, conjunctions, disjunctions, internal negations}	

Example	Conjunctive Normal Form
$(A \vee B) \wedge (C \vee D)$	Yes
$(A \vee \neg B) \wedge C \wedge (\neg D \vee E)$	Yes
$(A \vee \neg B) \wedge C \vee (\neg D \vee E)$	No
$(A \rightarrow \neg B) \wedge \neg(D \vee E)$	No

CNF Rules - Distribution Changes

Change a conjunction into a disjunction and vice-versa
Flip centre connective and distribute left predicate literals with right predicate

Example	Distribution Change
$(A \land B) \lor C$	$(A \lor C) \land (B \lor C)$
$(A \lor B) \land C$	$(A \land C) \lor (B \land C)$

CNF Rules – Negation Changes (De Morgan)

Moving negation inwards	$\neg(\forall x\ p) \equiv\ \exists x\ \neg p$
When the negation is applied, universal/existential quantifiers and equal/not equal signs are flipped	$\neg(\exists x\ p) \equiv\ \forall x\ \neg p$
	$\neg(likes(x,y) \wedge (y \neq z)) \equiv \neg likes(x,y) \vee y = z$
	$\neg(A \vee B) \equiv\ \neg A \wedge \neg B$
	$\neg(A \wedge B) \equiv\ \neg A \vee \neg B$

Double negation	$\neg\neg A \equiv\ A$

$\neg(y \neq z) \equiv (y = z)$

CNF Rules - Implication

$A \rightarrow B$ is equivalent to $\neg A \vee B$

A	B	$A \rightarrow B$	$\neg A \vee B$
T	T	T	T
T	F	F	F
F	T	T	T
F	F	T	T

CNF Rules – Bi-implication

$A \leftrightarrow B$ is equivalent to $(A \rightarrow B) \land (B \rightarrow A)$

A	B	$A \leftrightarrow B$	$(A \rightarrow B) \land (B \rightarrow A)$
T	T	T	T
T	F	F	F
F	T	F	F
F	F	T	T

CNF Conversion steps

1	Eliminate all bi-implication and implications
2	Move all negations inwards and correct double negations

CNF Conversion steps

3	Rename variables to ensure each quantifier has a unique one
	No convention with the naming, just choose a different letter. Adding a subscript number is fine as well, $x \approx x_0$

Examples	$\forall x\, P(x) \lor \exists x\, Q(x)$	
	$\approx$	$\forall x\, P(x) \lor \exists y\, Q(y)$

$\forall x\, A(x) \land \exists x\, B(x) \land \forall x\, C(x)$	
$\approx$	$\forall x_0\, A(x_0) \land \exists x_1\, B(x_1) \land \forall x_2\, C(x_2)$

CNF Conversion steps

4	Replace each existential quantifier (∃) with a Skolem constant
	Basically remove the quantifier and wrap its variable in some arbitrary predicate

Case 1	Existential quantifier is at the outermost level (no parent)
	Replace all occurrence of the variable with an arbitrary constant

Examples	$\exists x \ Q(x)$	
	$\approx$	$Q(s_0)$

CNF Conversion steps

4	Replace each existential quantifier (∃) with a Skolem constant
	Basically remove the quantifier and wrap its variable in some arbitrary predicate

Case 2	Existential quantifier is wrapped in a Universal quantifier
	Replace all occurrence of the variable with a predicate whose parameters include the universal quantifier variables

CNF Conversion steps

Case 2	Existential quantifier is wrapped in a Universal quantifier

Examples	$\forall x\ \exists y\ P(x, y)$	
	$\approx$	$\forall x\ P(x, S_0(x))$

$\forall x_0 \ldots \forall x_n\ \exists y\ P(x_0, \ldots, x_n, y)$	
$\approx$	$\forall x_0 \ldots \forall x_n\ P(x, S_0(x_0, \ldots, x_n))$

CNF Conversion steps

5	Drop all universal quantifiers
	Basically just remove them all, no other changes are necessary

Examples	$\forall x\ Person(x)$	
	$\approx$	$Person(x)$

CNF Conversion steps

6	Arrange the ∧ and ∨ connective to match conjunctive normal form
	Just used the basic CNF rules in the previous slides, nothing special is required from this point onwards

Question 3i

$\forall x\ (bird(x) \rightarrow flies(x))$

Step	Modification
1	$\forall x\ (\neg bird(x) \vee flies(x))$
2	N/A
3	N/A
4	N/A
5	$\neg bird(x) \vee flies(x)$
6	N/A
F	$\neg bird(x) \vee flies(x)$

Question 3ii

$\exists x\, \forall y\, \forall z\, (person(x) \wedge ((likes(x,y) \wedge y \neq z) \rightarrow \neg likes(x,z)))$

Step	Modification
1	$\exists x\, \forall y\, \forall z\, (person(x) \wedge (\neg(likes(x,y) \wedge y \neq z) \vee \neg likes(x,z)))$
2	$\exists x\, \forall y\, \forall z\, (person(x) \wedge (\neg likes(x,y) \vee y = z \vee \neg likes(x,z)))$
3	N/A
4	$\forall y\, \forall z\, (person(s_0) \wedge (\neg likes(s_0,y) \vee y = z \vee \neg likes(s_0,z)))$
5	$person(s_0) \wedge (\neg likes(s_0,y) \vee y = z \vee \neg likes(s_0,z))$
6	N/A
F	$person(s_0) \wedge (\neg likes(s_0,y) \vee y = z \vee \neg likes(s_0,z))$

Background – Resolution Rule

Create a new clause from two other clauses containing complementary literals

Clause 1	Clause 2	New Clause
$\neg Q \vee P$	$Q \vee P$	P
$\neg Q \vee R$	$\neg R$	$\neg Q$

Background – Resolution Rule

Create a new clause from two other clauses containing complementary literals

Clause 1	Clause 2	New Clause
{¬Q, P}	{Q, P}	{P}
{Q, R}	{¬R}	{¬Q}

Background – Resolution Refutation Steps

Determine what predicates in the clauses refute the proof

1. Negate the derivation
2. Convert all clauses to conjunctive normal form (CNF)
   - Including the derivation
3. Continually apply the resolution rule
   - Basically just combining clauses until all predicates are cancelled

Original Resolution Example

$(P \vee Q), P \rightarrow R, Q \rightarrow R \vdash R$

#	Original	Operation	Result
1	R	Conclusion Negation	$\neg R$
2	$(P \vee Q)$	CNF Conversion	$P \vee Q$
3	$P \rightarrow R$	CNF Conversion	$\neg P \vee R$
4	$Q \rightarrow R$	CNF Conversion	$\neg Q \vee R$

$\{\neg R, P \vee Q, \neg P \vee R, \neg Q \vee R\}$

Original Resolution Example

(P V Q), P → R, Q → R ⊢ R

#	Original	Operation	Result
5	P V Q, ¬P V R	Resolution rule 2, 3	Q V R
6	¬R, ¬P V R	Resolution rule 1, 3	¬P
7	¬R, ¬Q V R	Resolution rule 1, 4	¬Q
8	Q V R, ¬Q	Resolution rule, 5, 7	R
9	¬R, R	Resolution rule, 1, 8	.

{¬R, P V Q, ¬P V R, ¬Q V R}

Background – Resolution Refutation Differences

1	Rename variables to ensure they are unique amongst CNF clauses
	Once all clauses have been converted to CNF based on their quantifiers

Example	$\{\neg A(x) \vee B(x),\ \neg C(x) \vee D(x),\ E(x)\}$	
	$\approx$	$\{\neg A(x) \vee B(x),\ \neg C(y) \vee D(y),\ E(z)\}$

Background – Resolution Refutation Differences

2	Variables can be substituted to resolve their predicates

Example	$\neg A(x) \lor B(x), A(y)$		
	$\approx$	$\{x/y\}$	$B(y)$

Question 4v

$\forall x\,(P(x) \rightarrow Q(x)), \forall x(Q(x) \rightarrow R(x)) \vdash \forall x\,(P(x) \rightarrow R(x))$

#	Original	Operation	Result
1	$\forall x\,(P(x) \rightarrow R(x))$	Conclusion Negation	$\neg\forall x\,(P(x) \rightarrow R(x))$
2	$\forall x\,(P(x) \rightarrow Q(x))$	CNF Conversion	$\neg P(x) \lor Q(x)$
3	$\forall x(Q(x) \rightarrow R(x))$	CNF Conversion	$\neg Q(y) \lor R(y)$
	$\neg\forall x\,(P(x) \rightarrow R(x))$	CNF Conversion	$P(s_0) \land \neg R(s_0)$
4			$P(s_0)$
5			$\neg R(s_0)$

Question 4v

$$\forall x\,(P(x) \rightarrow Q(x)), \forall x(Q(x) \rightarrow R(x)) \vdash \forall x\,(P(x) \rightarrow R(x))$$

#	Original	Operation	Result
5	$\neg P(x) \vee Q(x), \neg Q(x) \vee R(x)$	Resolution 2, 3 $\{x/y\}$	$\neg P(y) \vee R(y)$
6	$\neg P(y) \vee R(y), P(s_0)$	Resolution 4, 5 $\{y/s_0\}$	$R(s_0)$
7	$R(s_0), \neg R(s_0)$	Resolution 6, 7	.

$$\{\neg P(x) \vee Q(x),\ \neg Q(y) \vee R(y),\ P(s_0),\ \neg R(s_0)\}$$

Question 5i

1	There is a computer scientist who likes every operating system
	$\exists x\ (cs(x) \land \forall y\ (os(y) \rightarrow likes(x, y))$

2	Linux is an operating system
	$os(Linux)$

3	Someone likes Linux
	$\exists z\ likes(z, Linux)$

Question 5ii

Basically convert to CNF

1	$\exists x\ (cs(x) \land \forall y\ (os(y) \rightarrow likes(x,y))$
	$cs(s_0) \land (\neg os(y) \lor likes(s_0,y))$
2	$os(Linux)$
	$os(Linux)$
3	$\exists z\ likes(z,Linux)$
	$\neg likes(z,Linux)$

Question 5iii

$A, B, \neg C$

#	Original	Operation	Result
$cs(s_0) \wedge (\neg os(y) \vee likes(s_0, y))$			
1			$cs(s_0)$
2			$\neg os(y) \vee likes(s_0, y)$
3			$os(Linux)$
4			$\neg likes(z, Linux)$
5	$\neg os(y) \vee likes(s_0, y), os(Linux)$	Resolution 2, 3 $\{y/Linux\}$	$likes(s_0, Linux)$
6	$\neg likes(z, Linux), likes(s_0, Linux)$	Resolution 4, 5 $\{z/s_0\}$	.

Background – Horn Clauses

Disjunction of literals containing at most a single positive literal	
Case 1	All literals are negated
Case 2	Exactly one literal is not negated

Basic idea is that the positive literal is implied by reversing the implication rule		
$\neg a \vee \neg b \vee \neg c \vee \neg d \vee \neg e \vee z$	$\approx$	$a \vee b \vee c \vee d \vee e \rightarrow z$

Question 5iv

All clauses from 5ii are Horn clauses

Even clause A, since the two disjunctions separated by the conjunction each have only one positive literal

Since all clauses are horn clauses and there exists a regular resolution for the empty clause, it is therefore possible to derive an SLD resolution for the empty clause as well.

Question 5iv – SLD Resolution

Target: $\neg likes(z, Linux)$

Previous clause is always used in next resolution

$[C_0, \ldots, C_n, C_t]$

$cs(s_0)$

$\neg os(y) \lor likes(s_0, y)$

$os(Linux)$

$\neg likes(z, Linux)$

$\neg likes(z, Linux)$

$\neg os(y)$

.

Question 5iv – SLD Resolution

Target:	$\neg likes(z, Linux)$

#	Original	Operation	Result
$cs(s_0) \wedge (\neg os(y) \vee likes(s_0, y))$			
1			$cs(s_0)$
2			$\neg os(y) \vee likes(s_0, y)$
3			$os(Linux)$
4			$\neg likes(z, Linux)$
5	$\neg os(y) \vee likes(s_0, y), \neg likes(z, Linux)$	Resolution 2, 4 $\{z/s_0, y/Linux\}$	$\neg os(Linux)$
6	$os(Linux), \neg os(Linux)$	Resolution 3, 5	.

Question 5v

If a computer scientist likes every operating system, and Linux is an operating system then they will like Linux

Computer scientist like every OS, Linus is an OS ⊢ computer scientist like Linux
$\exists x\ (cs(x) \land \forall y\ (os(y) \rightarrow likes(x, y))), os(Linux) \vdash \exists z\ likes(z, Linux)$

Notes

- Assignment 2 packages are as follows:
  - Preprocessing toolkit:
    - https://www.nltk.org/
  - Modelling toolkit
    - https://scikit-learn.org/stable/
- First-order logic: Conversion to CNF
  - https://april.eecs.umich.edu/courses/eecs492_w10/wiki/images/6/6b/CNF_conversion.pdf