Lecture 8 First-Order Logic
Models and Reasoning with Clauses

COMP24412: Symbolic AI

Giles Reger

February 2019

Aim and Learning Outcomes

The aim of this lecture is to:

Explore what it means for an interpretation to be a model of a formula and see how we can reason with formulas in clausal form

Learning Outcomes

By the end of this lecture you will be able to:

1. Identify when an interpretation is a model of a formula
2. Give examples of why first-order logic is more expressive than Datalog or Prolog
3. Explain the meaning of the open and closed world assumptions in terms of interpretations
4. Recall the resolution rule and how it applies to sets of clauses to solve reasoning problems

Recap

Datalog: closed-world, function-free, rules, matching

Prolog: closed-world, functions, rules, unification

First-order logic: open-world, functions, formulas

Free Variables, Sentences and Closure (Recap)

The free variables of a formula f are those not captured by a quantifer. Otherwise they are bound variables.

What are the free variables in the following:

$$p(x, y) \leftrightarrow \exists z.(r(x, z) \wedge r(z, y))$$
$$(\forall x.\exists y.p(y))$$
$$(\forall x.p(x, y)) \wedge (\exists x.p(x, y))$$

Free Variables, Sentences and Closure (Recap)

The free variables of a formula f are those not captured by a quantifer. Otherwise they are bound variables.

What are the free variables in the following:

$$p(x,y) \leftrightarrow \exists z.(r(x,z) \wedge r(z,y)) \qquad \{x,y\}$$
$$(\forall x.\exists y.p(y))$$
$$(\forall x.p(x,y)) \wedge (\exists x.p(x,y))$$

Free Variables, Sentences and Closure (Recap)

The free variables of a formula f are those not captured by a quantifer. Otherwise they are bound variables.

What are the free variables in the following:

$p(x,y) \leftrightarrow \exists z.(r(x,z) \wedge r(z,y))$	$\{x,y\}$
$(\forall x.\exists y.p(y))$	
$(\forall x.p(x,y)) \wedge (\exists x.p(x,y))$	

Free Variables, Sentences and Closure (Recap)

The free variables of a formula f are those not captured by a quantifer. Otherwise they are bound variables.

What are the free variables in the following:

$p(x, y) \leftrightarrow \exists z.(r(x, z) \land r(z, y))$	$\{x, y\}$
$(\forall x.\exists y.p(y))$	$\{\}$
$(\forall x.p(x, y)) \land (\exists x.p(x, y))$	

Free Variables, Sentences and Closure (Recap)

The free variables of a formula f are those not captured by a quantifer. Otherwise they are bound variables.

What are the free variables in the following:

$p(x, y) \leftrightarrow \exists z.(r(x, z) \wedge r(z, y))$	$\{x, y\}$
$(\forall x.\exists y.p(y))$	$\{\}$
$(\forall x.p(x, y)) \wedge (\exists x.p(x, y))$	

Free Variables, Sentences and Closure (Recap)

The free variables of a formula f are those not captured by a quantifer. Otherwise they are bound variables.

What are the free variables in the following:

$p(x,y) \leftrightarrow \exists z.(r(x,z) \wedge r(z,y))$	$\{x, y\}$
$(\forall x.\exists y.p(y))$	$\{\}$
$(\forall x.p(x,y)) \wedge (\exists x.p(x,y))$	$\{y\}$

Free Variables, Sentences and Closure (Recap)

The free variables of a formula f are those not captured by a quantifer. Otherwise they are bound variables.

What are the free variables in the following:

$$
\begin{array}{ll}
p(x,y) \leftrightarrow \exists z.(r(x,z) \wedge r(z,y)) & \{x,y\} \\
(\forall x.\exists y.p(y)) & \{\} \\
(\forall x.p(x,y)) \wedge (\exists x.p(x,y)) & \{y\}
\end{array}
$$

If ϕ has free variables X we might write it $\phi[X]$.
We write $\phi[V]$ for the formula $\phi[X]$ where X is replaced by V.

Free Variables, Sentences and Closure (Recap)

The free variables of a formula f are those not captured by a quantifer. Otherwise they are bound variables.

What are the free variables in the following:

$p(x,y) \leftrightarrow \exists z.(r(x,z) \wedge r(z,y))$	$\{x,y\}$
$(\forall x.\exists y.p(y))$	$\{\}$
$(\forall x.p(x,y)) \wedge (\exists x.p(x,y))$	$\{y\}$

If ϕ has free variables X we might write it $\phi[X]$.
We write $\phi[V]$ for the formula $\phi[X]$ where X is replaced by V.

A formula is a sentence if it does not contain any free variables

The universal closure of $\phi[X]$ is $\forall X.\phi[X]$ (similarly for existential)

Interpretation (Recap)

Let $\langle \mathcal{D}, \mathcal{I} \rangle$ be a structure such that $\mathcal{I}$ is an interpretation over a non-empty (possibly infinite) domain $\mathcal{D}$.

The map $\mathcal{I}$ maps

- Every constant symbol to an element of $\mathcal{D}$
- Every function symbol of arity n to a function in $\mathcal{D}^n \to \mathcal{D}$
- Every proposition symbol to a truth value in $\mathbb{B}$
- Every predicate symbol of arity n to a function in $\mathcal{D}^n \to \mathbb{B}$

We lift interpretations to non-constant terms and atoms recursively as

$$\begin{array}{ccc} \mathcal{I}(f(t_1,\ldots,t_n)) & = & \mathcal{I}(f)(\mathcal{I}(t_1),\ldots,\mathcal{I}(t_n)) \\ \mathcal{I}(t_1 = t_2) & = & \mathcal{I}(t_1) = \mathcal{I}(t_2) \\ \mathcal{I}(p(t_1,\ldots,t_n)) & = & \mathcal{I}(p)(\mathcal{I}(t_1),\ldots,\mathcal{I}(t_n)) \end{array}$$

Interpretation of Formulas (Recap)

Finally we interpret formulas

$\mathcal{I}(\mathit{true})$	is always true
$\mathcal{I}(\neg\phi)$	*iff* $\mathcal{I}(\phi)$ is not true
$\mathcal{I}(\phi_1 \wedge \phi_2)$	*iff* both $\mathcal{I}(\phi_1)$ and $\mathcal{I}(\phi_2)$ are true
$\mathcal{I}(\forall x.\phi[x])$	*iff* for every $d \in \mathcal{D}$ we have that $\mathcal{I}(\phi[d])$ is true

Recall - a formula can have many consistent interpretations

Validity vs Satisfiability/Consistency (Formally)

An interpretation satisfies a sentence if the sentence evaluates to true in it. We say that the interpretation is a model of that sentence

Is this a Model?

$$p(a, b) \wedge p(b, a)$$

$$\begin{array}{rcl} \mathcal{I}(a) & = & 1 \\ \mathcal{I}(b) & = & 2 \\ \mathcal{I}(p) & = & \{(1,2),(2,1),(1,1)\} \end{array}$$

Is this a Model?

$$p(a, b) \wedge p(b, a)$$

$$\begin{array}{rcl}
\mathcal{I}(a) & = & 1 \\
\mathcal{I}(b) & = & 2 \\
\mathcal{I}(p) & = & \{(1,2),(2,1),(1,1)\}
\end{array}$$

Yes

Is this a Model?

$$p(a, b) \land p(b, a) \land \neg p(a, a)$$

$$\begin{array}{rcl} \mathcal{I}(a) & = & 1 \\ \mathcal{I}(b) & = & 2 \\ \mathcal{I}(p) & = & \{(1,2),(2,1),(1,1)\} \end{array}$$

Is this a Model?

$$p(a, b) \land p(b, a) \land \neg p(a, a)$$

$$\begin{array}{rcl} \mathcal{I}(a) & = & 1 \\ \mathcal{I}(b) & = & 2 \\ \mathcal{I}(p) & = & \{(1,2),(2,1),(1,1)\} \end{array}$$

No

Is this a Model?

$$\forall x. \forall y. p(x, y)$$

$$\begin{array}{rcl} \mathcal{I}(a) & = & 1 \\ \mathcal{I}(b) & = & 2 \\ \mathcal{I}(p) & = & \{(1,2),(2,1),(1,1)\} \end{array}$$

Is this a Model?

$$\forall x. \forall y. p(x, y)$$

$$\begin{array}{rcl} \mathcal{I}(a) & = & 1 \\ \mathcal{I}(b) & = & 2 \\ \mathcal{I}(p) & = & \{(1,2),(2,1),(1,1)\} \end{array}$$

$\mathcal{I}(\forall x.\phi[x])$ *iff* for every $d \in \mathcal{D}$ we have that $\mathcal{I}(\phi[d])$ is true

Is this a Model?

$$\forall x.\forall y.p(x, y)$$

$$\begin{array}{rcl}\mathcal{I}(a) & = & 1 \\ \mathcal{I}(b) & = & 2 \\ \mathcal{I}(p) & = & \{(1,2),(2,1),(1,1)\}\end{array}$$

$\mathcal{I}(\forall x.\forall y.p(x, y))$ *iff* for every $d \in \mathcal{D}$ we have that $\mathcal{I}(\forall y.p(d, y))$ is true

Is this a Model?

$$\forall x.\forall y.p(x, y)$$

$$\begin{array}{rcl} \mathcal{I}(a) & = & 1 \\ \mathcal{I}(b) & = & 2 \\ \mathcal{I}(p) & = & \{(1,2),(2,1),(1,1)\} \end{array}$$

$\mathcal{I}(\forall y.p(1, y))$ is true and $\mathcal{I}(\forall y.p(2, y))$ is true

Is this a Model?

$$\forall x.\forall y.p(x, y)$$

$$\begin{array}{rcl}\mathcal{I}(a) & = & 1 \\ \mathcal{I}(b) & = & 2 \\ \mathcal{I}(p) & = & \{(1,2),(2,1),(1,1)\}\end{array}$$

$\mathcal{I}(p(1,1))$ is true, $\mathcal{I}(p(1,2))$ is true, $\mathcal{I}(p(2,1))$ is true, $\mathcal{I}(p(2,2))$ is true

Is this a Model?

$$\forall x. \forall y. p(x, y)$$

$$\begin{array}{rcl} \mathcal{I}(a) & = & 1 \\ \mathcal{I}(b) & = & 2 \\ \mathcal{I}(p) & = & \{(1,2),(2,1),(1,1)\} \end{array}$$

$\mathcal{I}(p(1,1))$ is true, $\mathcal{I}(p(1,2))$ is true, $\mathcal{I}(p(2,1))$ is true, $\mathcal{I}(p(2,2))$ is true

Is this a Model?

$$\forall x.\forall y.p(x, y)$$

$$\begin{aligned}
\mathcal{I}(a) &= 1 \\
\mathcal{I}(b) &= 2 \\
\mathcal{I}(p) &= \{(1,2),(2,1),(1,1)\}
\end{aligned}$$

No

Is this a Model?

$$\exists x, y, z.(x \neq y \wedge x \neq z \wedge y \neq z)$$

$$\begin{array}{rcl} \mathcal{I}(a) & = & 1 \\ \mathcal{I}(b) & = & 2 \\ \mathcal{I}(p) & = & \{(1,2),(2,1),(1,1)\} \end{array}$$

Is this a Model?

$$\exists x, y, z.(x \neq y \wedge x \neq z \wedge y \neq z)$$

$$\begin{array}{rcl} \mathcal{I}(a) & = & 1 \\ \mathcal{I}(b) & = & 2 \\ \mathcal{I}(p) & = & \{(1,2),(2,1),(1,1)\} \end{array}$$

No

Is this a Model?

$$\exists x, y, z.(x \neq y \land x \neq z \land y \neq z)$$

$$\begin{array}{rcl}
\mathcal{I}(a) & = & 1 \\
\mathcal{I}(b) & = & 2 \\
\mathcal{I}(c) & = & 3 \\
\mathcal{I}(p) & = & \{(1,2),(2,1),(1,1)\}
\end{array}$$

Is this a Model?

$$\exists x, y, z.(x \neq y \wedge x \neq z \wedge y \neq z)$$

$$\begin{aligned}\mathcal{I}(a) &= 1\\ \mathcal{I}(b) &= 2\\ \mathcal{I}(c) &= 3\\ \mathcal{I}(p) &= \{(1,2),(2,1),(1,1)\}\end{aligned}$$

Yes

Is this a Model?

$$\forall x.(x \neq suc(x)) \land \forall x.\exists y.(suc(x) = y) \land \neg\exists x.(zero = suc(x))$$

$$\begin{array}{lcl} \mathcal{I}(zero) & = & 1 \\ \mathcal{I}(suc(1)) & = & 2 \end{array}$$

Is this a Model?

$$\forall x.(x \neq suc(x)) \land \forall x.\exists y.(suc(x) = y) \land \neg\exists x.(zero = suc(x))$$

$$\begin{array}{lcl} \mathcal{I}(zero) & = & 1 \\ \mathcal{I}(suc(1)) & = & 2 \end{array}$$

No

Is this a Model?

$$\forall x.(x \neq suc(x)) \land \forall x.\exists y.(suc(x) = y) \land \neg\exists x.(zero = suc(x))$$

$$\begin{array}{lcl} \mathcal{I}(zero) & = & 1 \\ \mathcal{I}(suc(1)) & = & 2 \\ \mathcal{I}(suc(2)) & = & 3 \\ \mathcal{I}(suc(3)) & = & 4 \end{array}$$

Is this a Model?

$$\forall x.(x \neq suc(x)) \wedge \forall x.\exists y.(suc(x) = y) \wedge \neg\exists x.(zero = suc(x))$$

$$\begin{array}{lcl} \mathcal{I}(zero) & = & 1 \\ \mathcal{I}(suc(1)) & = & 2 \\ \mathcal{I}(suc(2)) & = & 3 \\ \mathcal{I}(suc(3)) & = & 4 \end{array}$$

No

Is this a Model?

$$\forall x.(x \neq suc(x)) \wedge \forall x.\exists y.(suc(x) = y) \wedge \neg\exists x.(zero = suc(x))$$

$$\begin{array}{lcl}
\mathcal{I}(zero) & = & 1 \\
\mathcal{I}(suc(1)) & = & 2 \\
\mathcal{I}(suc(2)) & = & 3 \\
\mathcal{I}(suc(3)) & = & 2
\end{array}$$

Is this a Model?

$$\forall x.(x \neq suc(x)) \land \forall x.\exists y.(suc(x) = y) \land \neg\exists x.(zero = suc(x))$$

$$\begin{array}{lcl} \mathcal{I}(zero) & = & 1 \\ \mathcal{I}(suc(1)) & = & 2 \\ \mathcal{I}(suc(2)) & = & 3 \\ \mathcal{I}(suc(3)) & = & 2 \end{array}$$

Yes

Is this a Model?

$$\forall x.(x \neq suc(x)) \land \forall x.\exists y.(suc(x) = y) \land \neg\exists x.(zero = suc(x))$$

$$\forall x, y.(suc(x) = suc(y) \rightarrow x = y)$$

$$\begin{array}{lcl} \mathcal{I}(zero) & = & 1 \\ \mathcal{I}(suc(1)) & = & 2 \\ \mathcal{I}(suc(2)) & = & 3 \\ \mathcal{I}(suc(3)) & = & 2 \end{array}$$

Is this a Model?

$$\forall x.(x \neq suc(x)) \wedge \forall x.\exists y.(suc(x) = y) \wedge \neg\exists x.(zero = suc(x))$$

$$\forall x, y.(suc(x) = suc(y) \rightarrow x = y)$$

$$\begin{array}{lcl} \mathcal{I}(zero) & = & 1 \\ \mathcal{I}(suc(1)) & = & 2 \\ \mathcal{I}(suc(2)) & = & 3 \\ \mathcal{I}(suc(3)) & = & 2 \end{array}$$

No

Is this a Model?

$$\forall x.(x \neq suc(x)) \wedge \forall x.\exists y.(suc(x) = y) \wedge \neg\exists x.(zero = suc(x))$$
$$\forall x, y.(suc(x) = suc(y) \rightarrow x = y)$$

$$\begin{array}{lcl}
\mathcal{I}(zero) & = & 1 \\
\mathcal{I}(suc(1)) & = & 2 \\
\mathcal{I}(suc(2)) & = & 3 \\
\mathcal{I}(suc(3)) & = & 4 \\
\mathcal{I}(suc(4)) & = & 5 \\
\ldots & &
\end{array}$$

Is this a Model?

$$\forall x.(x \neq suc(x)) \wedge \forall x.\exists y.(suc(x) = y) \wedge \neg\exists x.(zero = suc(x))$$
$$\forall x, y.(suc(x) = suc(y) \rightarrow x = y)$$

$$\begin{array}{lcl} \mathcal{I}(zero) & = & 1 \\ \mathcal{I}(suc(1)) & = & 2 \\ \mathcal{I}(suc(2)) & = & 3 \\ \mathcal{I}(suc(3)) & = & 4 \\ \mathcal{I}(suc(4)) & = & 5 \\ \ldots & & \end{array}$$

Yes, it's infinite

Validity vs Satisfiability/Consistency (Formally)

An interpretation satisfies a sentence if the sentence evaluates to true in it. We say that the interpretation is a model of that sentence

Validity vs Satisfiability/Consistency (Formally)

An interpretation satisfies a sentence if the sentence evaluates to true in it. We say that the interpretation is a model of that sentence

An interpretation satisfies a formula if it satisfies its universal closure.

A formula is satisfiable or consistent if it has a model.

Validity vs Satisfiability/Consistency (Formally)

An interpretation satisfies a sentence if the sentence evaluates to true in it. We say that the interpretation is a model of that sentence

An interpretation satisfies a formula if it satisfies its universal closure.

A formula is satisfiable or consistent if it has a model.

A formula is valid if *every* interpretation satisfies it.

Validity vs Satisfiability/Consistency (Formally)

An interpretation satisfies a sentence if the sentence evaluates to true in it. We say that the interpretation is a model of that sentence

An interpretation satisfies a formula if it satisfies its universal closure.

A formula is satisfiable or consistent if it has a model.

A formula is valid if *every* interpretation satisfies it.

A formula unsatisfiable or inconsistent if it has no models.

Validity vs Satisfiability/Consistency (Formally)

An interpretation satisfies a sentence if the sentence evaluates to true in it. We say that the interpretation is a model of that sentence

An interpretation satisfies a formula if it satisfies its universal closure.

A formula is satisfiable or consistent if it has a model.

A formula is valid if *every* interpretation satisfies it.

A formula unsatisfiable or inconsistent if it has no models.

Try writing down a consistent, a valid, and an inconsistent formula.

Entailment

We lift these notions to sets of formulas by interpreting them as conjunctions e.g. we can talk about a consistent set of formulas.

Entailment

We lift these notions to sets of formulas by interpreting them as conjunctions e.g. we can talk about a consistent set of formulas.

A formula ϕ is entailed by a set of formulas Γ if every model of Γ is also a model of ϕ, we write this

$$\Gamma \models \phi$$

also read as f is a consequence of Γ.

Entailment

We lift these notions to sets of formulas by interpreting them as conjunctions e.g. we can talk about a consistent set of formulas.

A formula ϕ is entailed by a set of formulas Γ if every model of Γ is also a model of ϕ, we write this

$$\Gamma \models \phi$$

also read as f is a consequence of Γ.

This is equivalent to $\Gamma \to \phi$ being valid.

Entailment

We lift these notions to sets of formulas by interpreting them as conjunctions e.g. we can talk about a consistent set of formulas.

A formula ϕ is entailed by a set of formulas Γ if every model of Γ is also a model of ϕ, we write this

$$\Gamma \models \phi$$

also read as f is a consequence of Γ.

This is equivalent to $\Gamma \rightarrow \phi$ being valid.

What if Γ is inconsistent?

Demonstrating Validity, Consistency, and Entailment

When there can be many interpretations the notion of truth can change.

Let us take an example

$$man(aristotle) \qquad human(cleopatra) \qquad \forall x.(man(x) \rightarrow human(x))$$

There exists a model where cleopatra is a man.

In every model aristotle is human.

Demonstrating Validity, Consistency, and Entailment

When there can be many interpretations the notion of truth can change.

Let us take an example

$$man(aristotle) \qquad human(cleopatra) \qquad \forall x.(man(x) \rightarrow human(x))$$

There exists a model where cleopatra is a man.

The set $\Gamma \cup man(cleopatra)$ is satisfiable or consistent

In every model aristotle is human.

Demonstrating Validity, Consistency, and Entailment

When there can be many interpretations the notion of truth can change.

Let us take an example

$$man(aristotle) \qquad human(cleopatra) \qquad \forall x.(man(x) \rightarrow human(x))$$

There exists a model where cleopatra is a man.

The set $\Gamma \cup man(cleopatra)$ is satisfiable or consistent

In every model aristotle is human.

The statement $man(aristotle)$ is entailed by Γ

Demonstrating Validity, Consistency, and Entailment

When there can be many interpretations the notion of truth can change.

Let us take an example

$$man(aristotle) \qquad human(cleopatra) \qquad \forall x.(man(x) \rightarrow human(x))$$

There exists a model where cleopatra is a man.

The set $\Gamma \cup man(cleopatra)$ is satisfiable or consistent

In every model aristotle is human.

The statement $man(aristotle)$ is entailed by Γ
The formula $\Gamma \rightarrow man(aristotle)$ is valid

Relation Between (In)Consistency and Validity

Let Γ be a set (conjunction) of formulas and ϕ be a sentence

The models of ϕ are exactly those that are not the models of $\neg\phi$

If Γ is inconsistent then $\Gamma \models$ *false*.

$\Gamma \models \phi$ if and only if $\Gamma \cup \{\neg\phi\} \models$ *false*

Relation Between (In)Consistency and Validity

Let Γ be a set (conjunction) of formulas and ϕ be a sentence

The models of ϕ are exactly those that are not the models of $\neg\phi$

If Γ is inconsistent then $\Gamma \models$ *false*.

If Γ is inconsistent then it has no models, *false* has no models, all models of Γ are also models of *false*.

$\Gamma \models \phi$ if and only if $\Gamma \cup \{\neg\phi\} \models$ *false*

Relation Between (In)Consistency and Validity

Let Γ be a set (conjunction) of formulas and ϕ be a sentence

The models of ϕ are exactly those that are not the models of $\neg\phi$

If Γ is inconsistent then $\Gamma \models \mathit{false}$.

If Γ is inconsistent then it has no models, *false* has no models, all models of Γ are also models of *false*.

$\Gamma \models \phi$ if and only if $\Gamma \cup \{\neg\phi\} \models \mathit{false}$

If ϕ is true in all models of Γ then $\neg\phi$ must be true in no models of Γ

No Database Semantics

Notice we have not assumed database semantics

Can model

- Domain Closure Assumption by explicitly referring to the domain and its closure e.g.

$$\forall x.(colour(x) \rightarrow (x = red \lor x = blue \lor \ldots))$$

- Unique Names Assumption by explicitly stating this, although this only works for explicitly named things. Due to the open world interpretation, not everything needs to be explicitly named.

The Closed World Assumption is difficult to model and attempts to do so are not very friendly.

Open vs Closed World

The closed world assumption forces the single interpretation where the minimum possible is true and everything else is false.

In an open world setting that minimal truth is still true but we do not constrain the truth of anything else.

Sometimes the former can be useful, sometimes it can be overly restrictive. It is important to know which setting you are working in.

Open vs Closed World

The closed world assumption forces the single interpretation where the minimum possible is true and everything else is false.

In an open world setting that minimal truth is still true but we do not constrain the truth of anything else.

Sometimes the former can be useful, sometimes it can be overly restrictive. It is important to know which setting you are working in.

Closed-world reasoning is generally non-monotonic i.e. if you learn new facts to be true then things that were previously true may become false (due to negation-as-failure).

Negation in an Open World

In an *open world* setting that minimal truth is still true but we do not constrain the truth of anything else.

Negation is used to restrict what can be true.

For example,

$$\forall x, y.((parent(x, y) \rightarrow \neg parent(y, x))$$

e.g. the *parent* relation is asymmetric.

Or simply

$$\neg loves(giles, marmite)$$

Knowledge Base Queries as Entailment

A (purely logical) knowledge base can be turned into a FOL formula by universally closing rules and conjoining all rules and facts.

A query can be turned into a FOL formula by existential closure e.g. query $\phi[X]$ becomes $\exists X.\phi[X]$

Given knowledge base Γ and a query ϕ both in FOL form, if the query is true then necessarily

$$\Gamma \models \phi$$

but the converse may not hold (if assuming database semantics).

Any Formula can be a 'Query' in FOL

In Prolog we are restricted by the kinds of queries we could ask.

```
car(X) :- hasWheels(X), hasEngine(X).
supercar(X) :- car(X), reallyFast(X).
```

Do all supercars have wheels?

Any Formula can be a 'Query' in FOL

In Prolog we are restricted by the kinds of queries we could ask.

```
car(X) :- hasWheels(X), hasEngine(X).
supercar(X) :- car(X), reallyFast(X).
```

Do all supercars have wheels?

Is the rule `supercar(X) :- hasWheels(X)` entailed?

Any Formula can be a ‘Query’ in FOL

In Prolog we are restricted by the kinds of queries we could ask.

```
car(X) :- hasWheels(X), hasEngine(X).
supercar(X) :- car(X), reallyFast(X).
```

Do all supercars have wheels?

Is the rule `supercar(X) :- hasWheels(X)` entailed?

$$\begin{array}{l} \forall x.((\mathit{wheels}(x) \land \mathit{engine}(x)) \rightarrow \mathit{car}(x)) \\ \forall x.((\mathit{car}(x) \land \mathit{fast}(x)) \rightarrow \mathit{supercar}(x)) \end{array} \models \forall x.(\mathit{wheels}(x) \rightarrow \mathit{supercar}(x))$$

Differences with Datalog/Prolog

Expressiveness:

- FOL can use negation
- FOL can use existential quantification
- FOL can have multiple facts in the head (e.g. pure disjunction)
- FOL allows arbitrary 'queries'

However:

- Prolog is a programming language with many non-logical parts. It is Turing-complete
- Prolog and Datalog have Database Semantics, which can be helpful from a modelling perspective

How do we reason in FOL?

We had forward chaining in Datalog and backward chaining in Prolog

Forward chaining worked by generating consequences

Backward chaining worked by subgoal reduction

We have similar parallels for FOL reasoning. I'm going to focus on forward-ish techniques

Reasoning with Implications

If someone is rich then they are happy
I am rich

Reasoning with Implications

If someone is rich then they are happy
I am rich

Therefore, I am happy

Reasoning with Implications

$rich \rightarrow happy$
$rich$

$happy$

Reasoning with Implications

$rich \rightarrow happy$
$rich$

$happy$

This is captured by the well-known Modus Ponens rule

$$\frac{A \rightarrow B \qquad A}{B}$$

It's what we were applying in forward chaining.

Reasoning with Implications

$rich \rightarrow happy$
$rich$

$happy$

This is captured by the well-known Modus Ponens rule

$$\frac{A \rightarrow B \qquad C}{B\theta} \quad \theta = \text{match}(A, C)$$

It's what we were applying in forward chaining. Actually this is.

Reasoning with Implications

$rich(X) \rightarrow happy(X)$
$rich(giles)$

$happy(giles)$

This is captured by the well-known Modus Ponens rule

$$\frac{A \rightarrow B \qquad C}{B\theta} \quad \theta = \text{match}(A, C)$$

It's what we were applying in forward chaining. Actually this is.

Modus Ponens

More generally it is written

$$\frac{A \rightarrow B \qquad C}{B\theta} \quad \theta = \text{mgu}(A, C)$$

where mgu stands for most general unifier - recall unification from Prolog.

A unifier is any unifying substitution and most general means any other unifier is a special case.

Modus Ponens

More generally it is written

$$\frac{A \rightarrow B \qquad C}{B\theta} \quad \theta = \mathrm{mgu}(A, C)$$

where mgu stands for most general unifier - recall unification from Prolog.

A unifier is any unifying substitution and most general means any other unifier is a special case.

Does this rule make sense from what we know about models?

Modus Ponens

More generally it is written

$$\frac{A \rightarrow B \qquad C}{B\theta} \quad \theta = \mathrm{mgu}(A, C)$$

where mgu stands for most general unifier - recall unification from Prolog.

A unifier is any unifying substitution and most general means any other unifier is a special case.

Does this rule make sense from what we know about models?

To be a sound inference every model of the premises should be a model of the conclusion

Modus Ponens

More generally it is written

$$\frac{A \rightarrow B \qquad A}{B}$$

where mgu stands for most general unifier - recall unification from Prolog.

A unifier is any unifying substitution and most general means any other unifier is a special case.

Does this rule make sense from what we know about models?

To be a sound inference every model of the premises should be a model of the conclusion

Modus Ponens

More generally it is written

$$\frac{A \rightarrow B \qquad C}{B\theta} \quad \theta = \mathrm{mgu}(A, C)$$

where mgu stands for most general unifier - recall unification from Prolog.

A unifier is any unifying substitution and most general means any other unifier is a special case.

Does this rule make sense from what we know about models?

To be a sound inference every model of the premises should be a model of the conclusion

Reasoning with Implications

If someone is rich then they are happy
I am rich or delusional

Therefore, ?

Reasoning with Implications

$rich(X) \rightarrow happy(X)$
$rich(giles) \vee delusional(giles)$

Therefore, ?

Reasoning with Implications

$rich(X) \rightarrow happy(X)$
$rich(giles) \vee delusional(giles)$

$happy(giles) \vee delusional(giles)$

Reasoning with Implications

$rich(X) \rightarrow happy(X)$
$rich(giles) \vee delusional(giles)$

$happy(giles) \vee delusional(giles)$

This is captured by generalisation of Modus Ponens called Resolution

$$\frac{A \rightarrow B \qquad A \vee D}{B \vee D}$$

Reasoning with Implications

$rich(X) \rightarrow happy(X)$
$rich(giles) \vee delusional(giles)$

$happy(giles) \vee delusional(giles)$

This is captured by generalisation of Modus Ponens called Resolution

$$\frac{A \rightarrow B \qquad C \vee D}{(B \vee D)\theta} \quad \theta = \text{mgu}(A, C)$$

Reasoning with Implications

$rich(X) \rightarrow happy(X)$
$rich(giles) \vee delusional(giles)$

$happy(giles) \vee delusional(giles)$

This is captured by generalisation of Modus Ponens called Resolution

$$\frac{\neg A \vee B \qquad C \vee D}{(B \vee D)\theta} \quad \theta = \mathrm{mgu}(A, C)$$

We resolve on A and C

Reasoning with Implications

$rich(X) \rightarrow happy(X)$
$rich(giles) \vee delusional(giles)$

$happy(giles) \vee delusional(giles)$

This is captured by generalisation of Modus Ponens called Resolution

$$\frac{\neg A \vee B \qquad C \vee D}{(B \vee D)\theta} \quad \theta = \mathrm{mgu}(A, C)$$

We resolve on A and C

$(\neg A \vee C)\theta$ must be valid as A and C unify

Clauses

A literal is an an atom or its negation. A clause is a disjunction of literals.

Clauses are implicitly universally quantified.

We can think of a clause as a conjunction implying a disjunction e.g.

$$(a_1 \wedge \ldots a_n) \rightarrow (b_1 \vee \ldots \vee b_m)$$

An empty clause is false.

If $m \leq 1$ then a clause is Horn - this is what we have in Prolog.

If $m = 1$ then a clause is definite - this is what we have in Datalog.

From now on we write t, s for terms, l for literals and C, D for clauses.

Resolution

Resolution works on clauses

$$\frac{l_1 \vee C \qquad \neg l_2 \vee D}{(C \vee D)\theta} \quad \theta = \mathrm{mgu}(l_1, l_2)$$

For example

$$\frac{p(a,x) \vee r(x) \qquad \neg r(f(y)) \vee p(y,b)}{p(a,f(y)) \vee p(y,b)}$$

Do these two clauses resolve?

$$s(x,a,x) \vee p(x,b) \qquad \neg s(b,y,c) \vee \neg p(f(b),b)$$

Resolution

Resolution works on clauses

$$\frac{l_1 \vee C \qquad \neg l_2 \vee D}{(C \vee D)\theta} \quad \theta = \mathrm{mgu}(l_1, l_2)$$

For example

$$\frac{p(a,x) \vee r(x) \qquad \neg r(f(y)) \vee p(y,b)}{p(a, f(y)) \vee p(y,b)}$$

Do these two clauses resolve?

$$\frac{s(x,a,x) \vee p(x,b) \qquad \neg s(b,y,c) \vee \neg p(f(b),b)}{s(f(b), a, f(b) \vee \neg s(b,y,c)}$$

Refutational Based Reasoning

We are going to look at a reasoning method that works by refutation.

Recall $\Gamma \models \phi$ if and only if $\Gamma \cup \{\neg\phi\} \models \mathit{false}$

If we want to show that ϕ is entailed by Γ we can show that $\Gamma \cup \{\neg\phi\}$ is inconsistent

Refutational Based Reasoning

We are going to look at a reasoning method that works by refutation.

Recall $\Gamma \models \phi$ if and only if $\Gamma \cup \{\neg\phi\} \models$ *false*

If we want to show that ϕ is entailed by Γ we can show that $\Gamma \cup \{\neg\phi\}$ is inconsistent

This is refutational based reasoning.

We will saturate $\Gamma \cup \{\neg\phi\}$ until there is nothing left to add or we have derived *false*.

If we do not find *false* then $\Gamma \not\models \phi$.

There are some caveats we will meet later.

Resolving to false

$$\begin{array}{l} \neg rich(x) \vee happy(x) \\ rich(giles) \end{array} \models happy(giles)$$

Resolving to false

$$\neg rich(x) \vee happy(x)$$
$$rich(giles)$$
$$\neg happy(giles)$$

Resolving to false

$\neg rich(x) \vee happy(x)$
$rich(giles)$
$\neg happy(giles)$

Resolving to false

$\neg rich(x) \vee happy(x)$
$rich(giles)$
$\neg happy(giles)$
$\neg rich(giles)$

Resolving to false

$\neg rich(x) \vee happy(x)$
$rich(giles)$
$\neg happy(giles)$
$\neg rich(giles)$

Resolving to false

$\neg rich(x) \vee happy(x)$
$rich(giles)$
$\neg happy(giles)$
$\neg rich(giles)$
$false$

Resolving to false

$\neg rich(x) \vee happy(x)$
$rich(giles)$
$\neg happy(giles)$
$\neg rich(giles)$
$false$

We could have done it in the other order (picked $\neg rich(x)$ first). We'll find out later that it's better to organise proof search to avoid this redundancy.

What about Equality?

$$\begin{array}{ll} \neg rich(father(x)) \vee happy(x) & \\ rich(david) & \models happy(giles) \\ father(giles) = david & \end{array}$$

What about Equality?

$$\neg rich(father(x)) \vee happy(x)$$
$$rich(david)$$
$$father(giles) = david$$
$$\neg happy(giles)$$

What about Equality?

$$\neg rich(father(x)) \vee happy(x)$$
$$rich(david)$$
$$father(giles) = david$$
$$\neg happy(giles)$$
$$\color{red}{\neg rich(father(giles))}$$

Paramodulation

The paramodulation rule lifts the idea behind resolution to equality

$$\frac{C \vee s = t \qquad l[u] \vee D}{(l[t] \vee C \vee D)\theta} \quad \theta = \mathrm{mgu}(s, u)$$

where u is not a variable.

For example

$$\frac{\mathit{father}(\mathit{giles}) = \mathit{david} \qquad \neg \mathit{rich}(\mathit{father}(x)) \vee \mathit{happy}(x)}{\neg \mathit{rich}(\mathit{david}) \vee \mathit{happy}(\mathit{giles})}$$

where $u = \mathit{father}(x)$.