

CS208 (Semester 1) Topic 8 : Semantics of Predicate Logic

Dr. Robert Atkey

Computer & Information Sciences

Semantics of Predicate Logic, Part 1

Models

So far: Syntax and Proof

1. Topic 3: The syntax of predicate logic
What sequences of symbols are well formed?
2. Topic 4: Proofs for predicate logic
When are formulas consequences of other formulas?
3. Topics 5-7: Hoare Logic
An Application of Predicate Logic

Missing so far: semantics

1. For Propositional Logic, we defined the *semantics* (“meaning”) of a formula P :
 - ▶ For every *valuation* v ,
the formula P is assigned a meaning $\llbracket P \rrbracket v$ which is either T or F.
2. This definition enabled us to give a definition of *entailment*:

$$P_1, \dots, P_n \models Q$$

which defines consequence without using proofs.

Semantics for Predicate Logic

The plan:

1. Fix *vocabularies* $\mathcal{V}$
2. Define *models* $\mathcal{M}$ for $\mathcal{V}$
3. Interpret *formulas* P using vocabulary $\mathcal{V}$ in models $\mathcal{M}$
4. Find uses for models (databases, generation, ...)
5. ...
6. Profit!

Fixing a Vocabulary

The function symbols we will use, and their *arities* (number of arguments):

Function name(s)	Arity
socrates	0
dayAfter	1
+, −	2

We write “func/*n*” for function symbol func with arity *n*

Fixing a Vocabulary

The predicates / relation symbols we will use, and their arities:

Predicate name(s)	Arity
human, mortal	1
$<$, $\leq$, $=$	2
between	3

We write “pred/n” for predicate symbol pred with arity n

A simplification

To keep things simple, I'm going to assume that we don't have any function symbols in our vocabulary.

Example: Orderings

- ▶ $\leq/2$ “less than”

Example: Places

- ▶ city/1 “is a city”
- ▶ within/2 “is within”
- ▶ country/1 “is a country”

Example: Forestry and Birdwatching

- ▶ tree/1 “is a tree”
- ▶ green/1 “is green”
- ▶ bird/1 “is a bird”
- ▶ satIn/2 “has sat in”

Models

With a fixed vocabulary, a *model* $\mathcal{M}$ is:

1. A *universe* $\mathcal{U}$, which is a set of individuals:

$$\mathcal{U} = \{1, 2, \text{socrates}, \text{hypatia}, \text{noether}, \text{alexandria}, \text{glasgow}, \dots\}$$

2. For each predicate pred/n , an n -ary relation on the set $\mathcal{U}$.

Relations

Several ways of understanding what a relation is:

1. For every n elements from $\mathcal{U}$, the interpretation of pred/n assigns the value T or F.
2. The interpretation of pred/n is a (possibly infinite) table of elements of $\mathcal{U}$ with n columns.
3. The interpretation of pred/n is a subset of the n -fold *cartesian product* $\underbrace{\mathcal{U} \times \cdots \times \mathcal{U}}_{n \text{ times}}$.

Example: Places, interpretation 1

$U = \{\text{aberdeen, edinburgh, glasgow, scotland, birmingham, england}\}$

$\text{city} = \{\text{aberdeen, edinburgh, glasgow, birmingham}\}$

$\text{country} = \{\text{england, scotland}\}$

$\text{within} = \{(\text{aberdeen, scotland}), (\text{edinburgh, scotland}),$
 $(\text{glasgow, scotland}), (\text{birmingham, england})\}$

Example: Places, interpretation 1

$\mathcal{U} = \{\text{aberdeen, edinburgh, glasgow, scotland, birmingham, england}\}$

As tables:

city
aberdeen
edinburgh
glasgow
birmingham

country
england
scotland

within	
aberdeen	scotland
edinburgh	scotland
glasgow	scotland
birmingham	england

Example: Places, interpretation 2

$\mathcal{U} = \{\text{loopland}\}$

$\text{city} = \{\text{loopland}\}$

$\text{country} = \{\text{loopland}\}$

$\text{within} = \{(\text{loopland}, \text{loopland})\}$

Example: Places, interpretation 2

$\mathcal{U} = \{\text{loopland}\}$

$\text{city} = \{\text{loopland}\}$

$\text{country} = \{\text{loopland}\}$

$\text{within} = \{(\text{loopland}, \text{loopland})\}$

The names are only there to separate the predicates. Models do not have to match our intuition for the names.

Example: Interpreting ordering with natural numbers

1. $\mathcal{U} = \{0, 1, 2, \dots\} = \mathbb{N}$ (all positive whole numbers)
2. The interpretation of $\leq/2$ is all pairs (x, y) such that $x \leq y$

Example: Interpreting ordering with rational numbers

1. $U = \{0, -1, 1, -\frac{1}{2}, \frac{1}{2}, -2, 2, \dots\} = \mathbb{Q}$
2. The interpretation of $\leq/2$ is all pairs (x, y) such that $x \leq y$

Example: Interpreting ordering with a small set

1. $U = \{a, b, c\}$
2. The interpretation of $\leq/2$ is the set:

$$\{(a, b), (a, c)\}$$

Note! not necessarily what we might think of as $\leq$! Need to add axioms.

Important Points

Every model $\mathcal{M}$ has

1. a universe; and
2. a relation for each predicate symbol pred/n ,

but the domain can be empty, or the predicate symbols' interpretations may be empty!

The model needn't match our intuition about the symbols!

- ▶ Will assume formulas that will restrict the possible models.

Relationship to Valuations

If all our predicate symbols have arity 0 (take no arguments), then a model consists of:

1. A universe $\mathcal{U}$; and
2. An assignment of T or F to each predicate symbol $\text{pred}/0$.

Apart from the universe, this is the same as a *valuation* in Propositional Logic (Topic 0).

Summary

We interpret Predicate Logic formulas in a *model* $\mathcal{M}$.

- ▶ A universe $\mathcal{U}$ – the set of all “things”.
- ▶ A relation between elements of $\mathcal{U}$ for every predicate.

Useful intuition: models are (possibly infinite) databases.

Semantics of Predicate Logic, Part 2

Interpreting Formulas

Meaning of free variables

Assume a vocabulary $\mathcal{V}$ and model $\mathcal{M}$ are fixed.

Consider the formula:

$$\text{city}(x) \wedge \text{within}(x, y)$$

we can't give it a truth value until we know what x and y mean.

Cities Model

$U = \{\text{aberdeen, edinburgh, glasgow, scotland, birmingham, england}\}$

As tables:

city
aberdeen
edinburgh
glasgow
birmingham

country
england
scotland

within	
aberdeen	scotland
edinburgh	scotland
glasgow	scotland
birmingham	england

Meaning of free variables

With the cities model, if we set:

$x = \text{glasgow}$

$y = \text{scotland}$

then $\text{city}(x) \wedge \text{within}(x, y)$ should be assigned the truth value T.

Meaning of free variables

With the cities model, if we set:

$x = \text{glasgow}$

$y = \text{scotland}$

then $\text{city}(x) \wedge \text{within}(x, y)$ should be assigned the truth value T.

If we set:

$x = \text{scotland}$

$y = \text{edinburgh}$

then $\text{city}(x) \wedge \text{within}(x, y)$ should be assigned the truth value F.

Interpreting Formulas

If we fix:

1. a vocabulary $\mathcal{V}$;
2. a model $\mathcal{M}$ of that vocabulary;
3. an assignment v of elements of $\mathcal{U}$ to free variables of P .

then we can give a truth value $\llbracket P \rrbracket(\mathcal{M}, v)$ to P .

Interpreting Formulas

Relations:

$$\begin{aligned} \llbracket R(x_1, \dots, x_n) \rrbracket(\mathcal{M}, v) &= \text{T} && \text{if } (v(x_1), \dots, v(x_n)) \in R \\ &= \text{F} && \text{otherwise} \end{aligned}$$

$$\begin{aligned} \llbracket x = y \rrbracket(\mathcal{M}, v) &= \text{T} && \text{if } v(x) = v(y) \\ &= \text{F} && \text{otherwise} \end{aligned}$$

where R is one of the relations in $\mathcal{M}$.

Interpreting Formulas (Example)

With the cities model $\mathcal{M}$:

$$\llbracket \text{within}(x, y) \rrbracket(\mathcal{M}, [x \mapsto \text{edinburgh}, y \mapsto \text{scotland}]) = \text{T}$$

$$\llbracket \text{within}(x, y) \rrbracket(\mathcal{M}, [x \mapsto \text{edinburgh}, y \mapsto \text{england}]) = \text{F}$$

Interpreting Formulas

Quantifiers:

$$\begin{aligned}\llbracket \forall x.P \rrbracket(\mathcal{M}, v) &= \text{T} && \text{if for all } a \in \mathcal{U}, \llbracket P \rrbracket(\mathcal{M}, v[x \mapsto a]) = \text{T} \\ &= \text{F} && \text{otherwise}\end{aligned}$$

$$\begin{aligned}\llbracket \exists x.P \rrbracket(\mathcal{M}, v) &= \text{T} && \text{if exists } a \in \mathcal{U}, \text{ with } \llbracket P \rrbracket(\mathcal{M}, v[x \mapsto a]) = \text{T} \\ &= \text{F} && \text{otherwise}\end{aligned}$$

Notation $v[x \mapsto a]$ means the assignment that maps x to a and any other variable to whatever v mapped it to.

Interpreting Formulas (Example)

$$\llbracket \forall x. \text{city}(x) \rrbracket (\mathcal{M}, []) = F$$

because all of the following would need to be T:

$$\llbracket \text{city}(x) \rrbracket (\mathcal{M}, [x \mapsto \text{aberdeen}]) = T$$

$$\llbracket \text{city}(x) \rrbracket (\mathcal{M}, [x \mapsto \text{edinburgh}]) = T$$

$$\llbracket \text{city}(x) \rrbracket (\mathcal{M}, [x \mapsto \text{glasgow}]) = T$$

$$\llbracket \text{city}(x) \rrbracket (\mathcal{M}, [x \mapsto \text{birmingham}]) = T$$

$$\llbracket \text{city}(x) \rrbracket (\mathcal{M}, [x \mapsto \text{scotland}]) = F$$

$$\llbracket \text{city}(x) \rrbracket (\mathcal{M}, [x \mapsto \text{england}]) = F$$

Interpreting Formulas (Example)

$$\llbracket \exists x. \text{city}(x) \rrbracket (\mathcal{M}, []) = \text{T}$$

because only one of the following needs to be T:

$$\llbracket \text{city}(x) \rrbracket (\mathcal{M}, [x \mapsto \text{aberdeen}]) = \text{T}$$

$$\llbracket \text{city}(x) \rrbracket (\mathcal{M}, [x \mapsto \text{edinburgh}]) = \text{T}$$

$$\llbracket \text{city}(x) \rrbracket (\mathcal{M}, [x \mapsto \text{glasgow}]) = \text{T}$$

$$\llbracket \text{city}(x) \rrbracket (\mathcal{M}, [x \mapsto \text{birmingham}]) = \text{T}$$

$$\llbracket \text{city}(x) \rrbracket (\mathcal{M}, [x \mapsto \text{scotland}]) = \text{F}$$

$$\llbracket \text{city}(x) \rrbracket (\mathcal{M}, [x \mapsto \text{england}]) = \text{F}$$

Interpreting Formulas

Propositional Connectives:

$$\llbracket P \wedge Q \rrbracket(\mathcal{M}, v) = \llbracket P \rrbracket(\mathcal{M}, v) \wedge \llbracket Q \rrbracket(\mathcal{M}, v)$$

$$\llbracket P \vee Q \rrbracket(\mathcal{M}, v) = \llbracket P \rrbracket(\mathcal{M}, v) \vee \llbracket Q \rrbracket(\mathcal{M}, v)$$

$$\llbracket P \rightarrow Q \rrbracket(\mathcal{M}, v) = \llbracket P \rrbracket(\mathcal{M}, v) \rightarrow \llbracket Q \rrbracket(\mathcal{M}, v)$$

$$\llbracket \neg P \rrbracket(\mathcal{M}, v) = \neg \llbracket P \rrbracket(\mathcal{M}, v)$$

Interpreting Formulas (Example)

$$\llbracket \text{city}(x) \wedge \text{within}(x, y) \rrbracket (\mathcal{M}, [x \mapsto \text{edinburgh}, y \mapsto \text{scotland}]) = \text{T}$$

and

$$\llbracket \text{city}(x) \wedge \text{within}(x, y) \rrbracket (\mathcal{M}, [x \mapsto \text{edinburgh}, y \mapsto \text{birmingham}]) = \text{F}$$

and

$$\llbracket \text{city}(x) \vee \text{within}(x, y) \rrbracket (\mathcal{M}, [x \mapsto \text{edinburgh}, y \mapsto \text{birmingham}]) = \text{T}$$

Some notation

We write

$$\mathcal{M} \models P$$

when

$$\llbracket P \rrbracket(\mathcal{M}, []) = \top$$

meaning that $\mathcal{M}$ *is a model of* P .

Some notation

We write

$$\mathcal{M} \models P$$

when

$$\llbracket P \rrbracket(\mathcal{M}, []) = \top$$

meaning that $\mathcal{M}$ *is a model of* P .

We write $\mathcal{M} \not\models P$ if this is not the case.

Examples

If $\mathcal{M}$ is the cities model, then

$$\mathcal{M} \models \exists x.\text{city}(x) \quad (\text{there exists a city})$$

$$\mathcal{M} \not\models \forall x.\text{city}(x) \quad (\text{not everything is a city})$$

$$\mathcal{M} \models \forall x.\text{city}(x) \rightarrow (\exists y.\text{within}(x, y))$$

(every city is within something)

Entailment

Relative to a model $\mathcal{M}$:

$$\mathcal{M}; P_1, \dots, P_n \models Q$$

exactly when:

if all $\llbracket P_i \rrbracket(\mathcal{M}, \square) = \text{T}$, then $\llbracket Q \rrbracket(\mathcal{M}, \square) = \text{T}$.

If all the assumptions are true, then the conclusion must be true

Entailment

$$P_1, \dots, P_n \models Q$$

exactly when *for all* $\mathcal{M}$, we have $\mathcal{M}; P_1, \dots, P_n \models Q$.

Checking this is infeasible (at least naively): there are infinitely many models, and the models themselves may be infinite.

Which is one reason to use proof.

Provability and Soundness

Recall that

$$P_1, \dots, P_n \vdash Q$$

means that Q is provable from the assumptions $P_1, \dots, P_n$.

Provability and Soundness

Recall that

$$P_1, \dots, P_n \vdash Q$$

means that Q is provable from the assumptions $P_1, \dots, P_n$.

The proof system we have used so far is *sound*:

$$P_1, \dots, P_n \vdash Q \quad \text{implies} \quad P_1, \dots, P_n \models Q$$

- ▶ If it is provable, then it is true in all models!
- ▶ Using proof means that we do not need to check infinitely many models, some of which may be infinite themselves.

Completeness

If we add excluded middle, then the proof system is also *complete*:

$$P_1, \dots, P_n \models Q \quad \text{implies} \quad P_1, \dots, P_n \vdash Q$$

every true entailment has a proof

Completeness

If we add excluded middle, then the proof system is also *complete*:

$$P_1, \dots, P_n \models Q \quad \text{implies} \quad P_1, \dots, P_n \vdash Q$$

every true entailment has a proof

- ▶ This is a remarkable and non-trivial fact!
- ▶ First proved by Kurt Gödel: “Gödel’s Completeness Theorem”
- ▶ It also holds if we have infinitely many assumptions $P_1, \dots$, which means (because a proof can only use finitely many assumptions) that an entailment only depends on finitely many assumptions: Predicate logic is “compact”.

Summary

We have defined what it means for a Predicate Logic P formula to be true in some model $\mathcal{M}$, and entailment in all models.

- ▶ Just as with Propositional Logic, this is done by breaking down the formula into its constituent parts
- ▶ Must ensure that all free variables have an interpretation.
- ▶ When a formula is true in some model, we write $\mathcal{M} \models P$.
- ▶ Entailment ($P_1, \dots, P_n \models Q$) is defined with respect to all models.
- ▶ Our proof system is sound and (with excluded middle) complete for this definition of entailment.

Semantics of Predicate Logic, Part 3

Using Models

Using Models

Some of the things we can do with models:

1. Fix $\mathcal{M}$. Does $\mathcal{M} \models P$?

Model checking

2. Fix $\mathcal{M}$. For what P does $\mathcal{M} \models P$?

What is true about *this* situation?

3. Fix $\mathcal{M}$. For what $a_1, \dots, a_n$ does $\mathcal{M} \models P[a_1, \dots, a_n]$?

Database queries

4. Fix formulas $P_1, \dots, P_n$. Is there a model for them?

Generating *worlds* or *counterexamples*

Using Models: Model Checking

Given a model $\mathcal{M}$, computing $\mathcal{M} \models P$ works by breaking down the structure of P . For example:

$$\begin{aligned} & \text{Cities} \models \exists x. \text{city}(x) \wedge \text{country}(x) \\ = & \llbracket \exists x. \text{city}(x) \wedge \text{country}(x) \rrbracket(\mathcal{M}, []) \\ = & \text{some } a. \llbracket \text{city}(x) \wedge \text{country}(x) \rrbracket(\mathcal{M}, [x \mapsto a]) \\ = & \text{some } a. \llbracket \text{city}(x) \rrbracket(\mathcal{M}, [x \mapsto a]) \text{ and } \llbracket \text{country}(x) \rrbracket(\mathcal{M}, [x \mapsto a]) \\ = & \text{some } a. a \in \text{city} \text{ and } a \in \text{country} \\ = & \text{F} \quad (\text{because no such } a \text{ exists, by checking them all}) \end{aligned}$$

Using Models: Database Queries

The *Relational Model* is a system for organising data.

- ▶ Proposed by Edgar F. Codd in 1969.
- ▶ Implemented in PostgreSQL, MySQL, SQLite¹, MS SQL Server, Oracle, IBM DB2, ...
- ▶ Typically using the Structured Query Language (SQL)
- ▶ Built on a logical foundation:
 1. Vocabulary $\approx$ schema
 2. Predicates $\approx$ (finite) tables
 3. Formulas $\approx$ queries
 4. Models $\approx$ databases (collections of tables)

¹there are *at least* $\approx$ 1000 SQLite DBs in this room.

Using Models: Database Queries

An SQL query:

```
SELECT City.X FROM City, Within  
WHERE City.X = Within.X AND Within.Y = "scotland"
```

²terms and conditions apply

Using Models: Database Queries

An SQL query:

```
SELECT City.X FROM City, Within  
WHERE City.X = Within.X AND Within.Y = "scotland"
```

A formula, with free variable x :

$$\text{city}(x) \wedge \text{within}(x, \text{scotland})$$

The collection of x s that make this formula true in the model is the result of the query². The second half of CS209 introduces SQL.

²terms and conditions apply

Models for Formulas

Given a collection of formulas $P_1, \dots, P_n$, it is sometimes useful to generate a model $\mathcal{M}$ for them.

1. Sometimes it is easier to think about concrete examples
2. Models can be used as counterexamples *see later...*
3. Can be used to generate interesting things “*Generative AI*”

Models for Formulas

With the following formulas:

1. $\exists x. \exists y. \text{city}(x) \wedge \text{city}(y) \wedge x \neq y$
2. $\forall x. \text{city}(x) \rightarrow (\exists y. \text{country}(y) \wedge \text{within}(x, y))$
3. $\forall x. \neg(\text{city}(x) \wedge \text{country}(x))$

Models for Formulas

With the following formulas:

1. $\exists x. \exists y. \text{city}(x) \wedge \text{city}(y) \wedge x \neq y$
there exist at least two cities
2. $\forall x. \text{city}(x) \rightarrow (\exists y. \text{country}(y) \wedge \text{within}(x, y))$
3. $\forall x. \neg(\text{city}(x) \wedge \text{country}(x))$

Models for Formulas

With the following formulas:

1. $\exists x. \exists y. \text{city}(x) \wedge \text{city}(y) \wedge x \neq y$
there exist at least two cities
2. $\forall x. \text{city}(x) \rightarrow (\exists y. \text{country}(y) \wedge \text{within}(x, y))$
every city is in a country
3. $\forall x. \neg(\text{city}(x) \wedge \text{country}(x))$

Models for Formulas

With the following formulas:

1. $\exists x. \exists y. \text{city}(x) \wedge \text{city}(y) \wedge x \neq y$
there exist at least two cities
2. $\forall x. \text{city}(x) \rightarrow (\exists y. \text{country}(y) \wedge \text{within}(x, y))$
every city is in a country
3. $\forall x. \neg(\text{city}(x) \wedge \text{country}(x))$
nothing is both a city and a country

Models for Formulas

First attempt at a model (start minimal):

universe = $\{\}$

city = $\{\}$

country = $\{\}$

within = $\{\}$

Models for Formulas

First attempt at a model (start minimal):

universe = $\{\}$

city = $\{\}$

country = $\{\}$

within = $\{\}$

Does not satisfy $\exists x. \exists y. \text{city}(x) \wedge \text{city}(y) \wedge x \neq y$

Need at least two cities!

Models for Formulas

Second attempt at a model (the names are arbitrary!):

universe = {plockton, auchtermuchty}

city = {plockton, auchtermuchty}

country = {}

within = {}

Models for Formulas

Second attempt at a model (the names are arbitrary!):

universe = {plockton, auchtermuchty}

city = {plockton, auchtermuchty}

country = {}

within = {}

Satisfies $\exists x. \exists y. \text{city}(x) \wedge \text{city}(y) \wedge x \neq y$

Does not satisfy $\forall x. \text{city}(x) \rightarrow (\exists y. \text{country}(y) \wedge \text{within}(x, y))$

Every city needs a country!

Models for Formulas

Third attempt at a model:

universe = {plockton, auchtermuchty}

city = {plockton, auchtermuchty}

country = {plockton}

within = {(auchtermuchty, plockton)}

Models for Formulas

Third attempt at a model:

universe = {plockton, auchtermuchty}

city = {plockton, auchtermuchty}

country = {plockton}

within = {(auchtermuchty, plockton)}

Satisfies $\exists x. \exists y. \text{city}(x) \wedge \text{city}(y) \wedge x \neq y$

Does not satisfy $\forall x. \text{city}(x) \rightarrow (\exists y. \text{country}(y) \wedge \text{within}(x, y))$

plockton needs to be somewhere!

Models for Formulas

Fourth attempt at a model:

universe = {plockton, auchtermuchty}

city = {plockton, auchtermuchty}

country = {plockton}

within = {(auchtermuchty, plockton), (plockton, plockton)}

Models for Formulas

Fourth attempt at a model:

universe = {plockton, auchtermuchty}

city = {plockton, auchtermuchty}

country = {plockton}

within = {(auchtermuchty, plockton), (plockton, plockton)}

Satisfies $\exists x. \exists y. \text{city}(x) \wedge \text{city}(y) \wedge x \neq y$

Satisfies $\forall x. \text{city}(x) \rightarrow (\exists y. \text{country}(y) \wedge \text{within}(x, y))$

Does not satisfy $\forall x. \neg(\text{city}(x) \wedge \text{country}(x))$

Nothing can be a city and a country!

Models for Formulas

Fifth attempt at a model:

universe = {plockton, auchtermuchty, scotland}

city = {plockton, auchtermuchty}

country = {scotland}

within = {(auchtermuchty, plockton), (plockton, plockton),
(auchtermuchty, scotland), (plockton, scotland)}

Models for Formulas

Fifth attempt at a model:

universe = {plockton, auchtermuchty, scotland}

city = {plockton, auchtermuchty}

country = {scotland}

within = {(auchtermuchty, plockton), (plockton, plockton),
(auchtermuchty, scotland), (plockton, scotland)}

Satisfies $\exists x. \exists y. \text{city}(x) \wedge \text{city}(y) \wedge x \neq y$

Satisfies $\forall x. \text{city}(x) \rightarrow (\exists y. \text{country}(y) \wedge \text{within}(x, y))$

Satisfies $\forall x. \neg(\text{city}(x) \wedge \text{country}(x))$

Models for Formulas

Sixth attempt at a model:

universe = {plockton, auchtermuchty}

city = {plockton, auchtermuchty}

country = {scotland}

within = {(auchtermuchty, scotland), (plockton, scotland)}

Models for Formulas

Sixth attempt at a model:

universe = {plockton, auchtermuchty}

city = {plockton, auchtermuchty}

country = {scotland}

within = {(auchtermuchty, scotland), (plockton, scotland)}

Satisfies $\exists x. \exists y. \text{city}(x) \wedge \text{city}(y) \wedge x \neq y$

Satisfies $\forall x. \text{city}(x) \rightarrow (\exists y. \text{country}(y) \wedge \text{within}(x, y))$

Satisfies $\forall x. \neg(\text{city}(x) \wedge \text{country}(x))$

and doesn't have any extra stuff

Models for Formulas

General strategy for formulas $P_1, \dots, P_n$:

1. See what is forced to exist without conditions
(e.g., at least two cities)
2. See what is then forced to exist by these things existing
(e.g., every city needs a country)
3. Disjointness axioms force non-overlap and more existence
(e.g., not a city and a country)
4. Try to keep the model minimal
(without the first formula, the empty model would work!)

Proof and Counterexamples

Models can be used to show some formulas are *unprovable*.

Not the same as *failing to prove it!*

Proof and Counterexamples

Models can be used to show some formulas are *unprovable*.

Not the same as *failing to prove it!*

To show that there is no proof of $P_1, \dots, P_n \vdash Q$:

1. Find a model $\mathcal{M}_c$ that makes all of $\mathcal{M}_c \models P_1, \dots, P_n$, but $\mathcal{M}_c \not\models Q$. ($\mathcal{M}_c$ is the *countermodel*)
2. If we could prove $P_1, \dots, P_n \vdash Q$ then (by soundness):
 - ▶ every model $\mathcal{M}$ such that $\mathcal{M} \models P_1, \dots, P_n$, then $\mathcal{M} \models Q$
3. But we have $\mathcal{M}_c$ that supports $P_1, \dots, P_n$ and **not** Q
4. So $P_1, \dots, P_n \vdash Q$ is **not** provable.

Example

It is not possible to prove

$$\neg(\exists x.\text{country}(x))$$

from the assumptions:

1. $\exists x.\exists y.\text{city}(x) \wedge \text{city}(y) \wedge x \neq y$
2. $\forall x.\text{city}(x) \rightarrow (\exists y.\text{country}(y) \wedge \text{within}(x, y))$
3. $\forall x.\neg(\text{city}(x) \wedge \text{country}(x))$

Example

It is not possible to prove

$$\neg(\exists x.\text{country}(x))$$

from the assumptions:

1. $\exists x.\exists y.\text{city}(x) \wedge \text{city}(y) \wedge x \neq y$
2. $\forall x.\text{city}(x) \rightarrow (\exists y.\text{country}(y) \wedge \text{within}(x, y))$
3. $\forall x.\neg(\text{city}(x) \wedge \text{country}(x))$

The model above satisfies all these, but not $\neg(\exists x.\text{country}(x))$.

Therefore, there is no proof of this formula from these assumptions.

Summary

Several useful questions can be asked using models:

1. Model checking: does $\mathcal{M} \models P$?
2. Model checking for all values: for what x does $\mathcal{M} \models P[x]$?
database queries!
3. Model generation for formulas $P_1, \dots, P_n$
can be used to show unprovable things

With sufficiently adapted ideas of “formula” and “model”, many questions can be recast in terms of logic and implemented on a computer.