# Introduction to model theory

Model theory is concerned with attributing meaning to logical sentences

Basics of model theory

1. Interpretations in propositional logic

2. Model-theoretic notions of validity, logical consequence and satisfiability

3. Interpretations in $1^{st}$ order logic

4. Herbrand interpretations, Herbrand models for logic programs and minimal Herbrand models

5. Completion and fixed-point semantics

# Interpretations: propositional logic

**I**nterpretations are simply assignmnents of $TRUE$ ($t$) or $FALSE$ ($f$) to every proposition

– For e.g. given propositions $p$ and $q$, one possible interpretation assigns $p$ to $TRUE$ and $q$ to $FALSE$

– With this interpretation, other formulae may be true or false: $p \vee q$ is $TRUE$, and $p \wedge q$ is $FALSE$

**A**n interpretation that gives the value $TRUE$ for a formula is called a *model* for that formula

– Thus, $p = TRUE, q = FALSE$ is a model for $p \vee q$

# Models and validity

**T**here are at most $2^n$ interpretations with $n$ propositional variables

**N**ot all these may be models for a formula

| $p$ | $q$ | $p \leftarrow q$ | Model for $p \leftarrow q$? |
|---|---|---|---|
| $f$ | $f$ | $t$ | $\surd$ |
| $f$ | $t$ | $f$ | $\times$ |
| $t$ | $f$ | $t$ | $\surd$ |
| $t$ | $t$ | $t$ | $\surd$ |

**F**ormulae for which *every* interpretation is a model are said to be *valid*

| $p$ | $q$ | $(p \leftarrow q) \land q$ | $p \leftarrow (p \leftarrow q) \land q$ |
|---|---|---|---|
| $f$ | $f$ | $f$ | $t$ |
| $f$ | $t$ | $f$ | $t$ |
| $t$ | $f$ | $f$ | $t$ |
| $t$ | $t$ | $t$ | $t$ |

# Consequence and equivalence

Consider the formulae $p$ and $p \vee q$

- Every interpretation that makes $p$ true also makes $p \vee q$ true. That is, every model of $p$ is a model of $p \vee q$

If every model of a sentence (or formula) $s_1$ is also a model of a sentence $s_2$ then $s_2$ is said to be a *logical consequence* of $s_1$. Alternatively, $s_1$ *logically implies* $s_2$, or $s_1 \models s_2$

If every model of $s_1$ is a model of $s_2$ and every model of $s_2$ is a model of $s_1$ then $s_1$ and $s_2$ are logically equivalent, or $s_1 \equiv s_2$

- Verify $\sim (p \leftarrow q) \;\equiv\; q \wedge \sim p$

# Satisfiability and sets of sentences

**A** sentence is said to be *satisfiable* if it has at least 1 model. Otherwise it is said to be *unsatisfiable*

**A** set of sentences $S = \{s_1, \ldots, s_n\}$ is to be understood as the formula $s_1 \wedge \ldots \wedge s_n$

– Thus, an interpretation $I$ is a model for a set $S$ of sentences iff it is a model for every sentence $s_i$ in $S$

– A set of sentences $S$ is satisfiable iff the formula $s_1 \wedge \ldots \wedge s_n$ is satisfiable. That is, there is at least 1 interpretation that is a model for all of the $s_i$

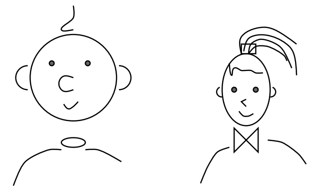
# Interpretations: $1^{st}$ order logic

These are more complicated as they require meanings for constants, functions and predicate symbols

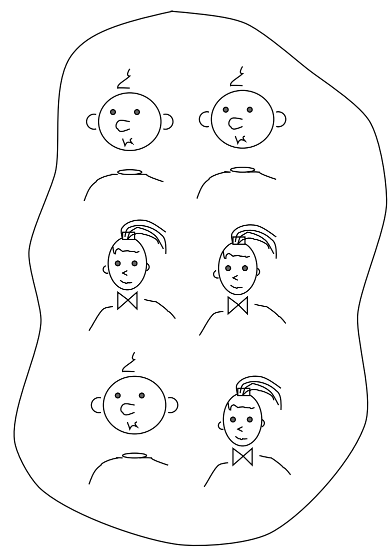
– For e.g. asking if $kesli(njoh, army)$ is true cannot be answered unless we know the meaning of each symbol. First, we have to state the objects in the domain of discourse:

The domain D

– Next, we have to map constants in our statement to objects in the domain




– Finally, the predicate $kesli/2$ has to be mapped to some relation that exists between objects in the domain




– We can now see that $kesli(njoh, army)$ is $TRUE$ as the objects corresponding to the ordered tuple $< njoh, army >$ are in the relation that $kesli$ represents

**A**n interpretation in $1^{st}$ order logic therefore requires specification of:

1. A domain $D$

2. A mapping of constants to elements in $D$

3. A mapping of $n$-ary function symbols to $n$-ary functions from $D^n \rightarrow D$

4. A mapping of $n$-ary predicate symbols to $n$-ary relations on $D^n$

– With these mappings, the statement $(\forall X)W$ is then $TRUE$ iff for every domain element that we can associate with $X$, $W$ is $TRUE$. $(\exists X)W$ is $TRUE$ iff for some domain element that we can associate with $X$, $W$ is $TRUE$

- The interpretation we have just provided makes $kesli(njoh, army)$ true. It is therefore a model for $kesli(njoh, army)$

- Changing any of the mappings 2-4 gives a different interpretation.

- Usually, when we write logic programs, we already have some interpretation in mind. This is often called the *intended interpretation*

# Herbrand interpretations and models

**I**nterpretations in $1^{st}$ order logic are more complex than propositional logic (as expected)

**Y**et logic programming systems appear to determine logical consequences without recourse to complex mappings

– Is an “intended interpretation” built-in?

– If so, will it work for any other interpretations?

**T**he logical consequence relation $P \models s$ requires that for *every* interpretation $I$, if $I$ is a model of $P$, then it is a model of $s$

**I**n fact, executing a logic program does not need to consider every interpretation. One special interpretation called the *Herbrand* interpretation is enough

**W**hy?

– A set of clauses $P$ has a model iff $P$ has a Herbrand interpretation that is a model (that is, a "Herbrand model")

– For definite-clause programs, there is a unique minimal Herbrand model

– For any definite-clause program $P$ and ground atom $s$, $P \models s$ iff $s$ is in the Herbrand model

# What are Herbrand interpretations?

**G**iven a program $P$ and a language $\mathcal{L}$ think of all ground terms that can be constructed

– For e.g. let $\mathcal{L}$ consist of the constant symbol 0, functions $s/1, p/1$ and predicate symbol $natural/1$. Let $P$ be:

$$natural(0) \leftarrow$$

$$natural(s(X)) \leftarrow natural(X)$$

– The set of all ground terms that can be constructed is the infinite set $\{0, s(0), p(0), s(p(0)), p(s(0)), \ldots\}$. This set is called the *Herbrand universe*

**N**ow think of all ground atoms that can be constructed using elements from the

Herbrand universe and predicate symbols in $P$

- Here, this is the infinite set $\{natural(0), natural(s(0)), \ldots\}$

- This is called the *Herbrand base* of $P$ or $\mathcal{B}(P)$

**A** Herbrand interpretation is simply an assignment of $TRUE$ to some subset of $\mathcal{B}(P)$ and $FALSE$ to the rest

- It is common to associate "Herbrand interpretation" only with the set of atoms assigned to $TRUE$

- Thus, $\{natural(0)\}$ is a Herbrand interpretation that assigns $TRUE$ to $natural(0)$ and $FALSE$ to everything else

# What are Herbrand models?

Consider the following program $P$:

$likes(john, X) \leftarrow likes(X, apples)$

$likes(mary, apples) \leftarrow$

Suppose the language $\mathcal{L}$ contained no symbols other than those in $P$.

– $\mathcal{B}(P)$ is the set $\{likes(john, john), likes(john, apples), likes(apples, john), likes(john, mary), likes(mary, john), likes(mary, apples), likes(apples, mary), likes(mary, mary), likes(apples, apples)\}$

– $\{likes(mary, apples), likes(john, mary)\}$ is a subset of $\mathcal{B}(P)$, and is a Herbrand interpretation

– It is a Herbrand model for $P$

- $\{likes(mary, apples), likes(john, mary), likes(mary, john)\}$ is also a model for $P$

# Ground instantiations and Herbrand models

**A** set of $1^{st}$ order clauses can be thought of as "short-hand" for a set of ground clauses

– The ground clauses are obtained by replacing variables by terms from the Herbrand universe (i.e. the set of all possible ground terms given $\mathcal{L}$).

– This is called the *ground instantiation* of $P$ or $\mathcal{G}(P)$.

– For e.g. $\mathcal{G}(P)$ for the earlier program:

$$likes(john, john) \leftarrow likes(john, apples)$$

$$likes(john, mary) \leftarrow likes(mary, apples)$$

$$likes(john, apples) \leftarrow likes(apples, apples)$$

$$likes(mary, apples) \leftarrow$$

– Now consider the earlier interpretation: $\{likes(mary, apples), likes(john, mary)\}$. Verify that this is a model for the $\mathcal{G}(P)$ above

**A** program $P$ has a model iff $\mathcal{G}(P)$ has a Herbrand model

# Models for definite-clauses

The set of all Herbrand models for a definite-clause program $P$ is partially ordered by $\subseteq$ and forms a lattice. For e.g.

For definite-clause programs, the minimal model is unique

The "meaning" of a definite-clause program is given by its minimal model

# Deduction theorem

Let $P = \{s_1, \ldots s_n\}$ be a set of clauses and $s$ be a sentence (not necessarily ground)

**Theorem.** $P \models s$ iff $P - \{s_i\} \models (s \leftarrow s_i)$

– Implication is preserved if we remove any sentence from the left and make it a condition on the right

$$P - s_1, \ldots, s_i \models (s \leftarrow s1 \wedge \ldots \wedge s_i)$$

$$\emptyset \models (s \leftarrow s1 \wedge \ldots \wedge s_n)$$

– That is, every model of $\emptyset$ is a model of $s \leftarrow s1 \wedge \ldots \wedge s_n$

– $s \leftarrow s1 \wedge \ldots \wedge s_n$ is valid

**N**ow consider $P \models q$

$p \leftarrow q \;\equiv\; \sim q \leftarrow \sim p$ and
$q \leftarrow \;\equiv\; q \leftarrow TRUE \;\equiv\; FALSE \leftarrow \sim q$

$P \models q \;\equiv\; P \models (q \leftarrow)$ iff:

$P \models (FALSE \leftarrow \sim q)$ iff:

$P \cup \{\sim q\} \models FALSE$

**T**hat is $P \models q$ iff $P \cup \{\sim q\}$ is unsatisfiable

Logical consequence can be checked by Refutation