Predicate Logic

- Invented by Gottlob Frege (1848–1925).
- Predicate Logic is also called “first-order logic”.

“Every good mathematician is at least half a philosopher, and every good philosopher is at least half a mathematician.”
Motivation

There are some kinds of human reasoning that we can’t do in propositional logic.

- For example:
  
  Every person likes ice cream.
  
  Billy is a person.
  
  Therefore, Billy likes ice cream.

- In propositional logic, the best we can do is $A \land B \rightarrow C$, which isn’t a tautology.
  
  - We’ve lost the internal structure.

We need to be able to refer to objects.

- We want to symbolize both a claim and the object about which the claim is made.
- We also need to refer to relations between objects,
  
  - as in “Waterloo is west of Toronto”.
- If we can refer to objects, we also want to be able to capture the meaning of every and some of.
- The predicates and quantifiers of predicate logic allow us to capture these concepts.
Apt-pet

- An apartment pet is a pet that is small
- Dog is a pet
- Cat is a pet
- Elephant is a pet
- Dogs and cats are small.
- Some dogs are cute
- Each dog hates some cat
- Fido is a dog

\[
\forall x \small(x) \land pet(x) \Rightarrow aptPet(x) \\
\forall x \dog(x) \Rightarrow pet(x) \\
\forall x \cat(x) \Rightarrow pet(x) \\
\forall x \elephant(x) \Rightarrow pet(x) \\
\forall x \dog(x) \Rightarrow small(x) \\
\forall x \cat(x) \Rightarrow small(x) \\
\exists x \dog(x) \land cute(x) \\
\forall x \dog(x) \Rightarrow \exists y \cat(y) \land hates(x, y) \\
dog(fido)
\]

Quantifiers

- Universal quantification (\(\forall\)) corresponds to finite or infinite conjunction of the application of the predicate to all elements of the domain.
- Existential quantification (\(\exists\)) corresponds to finite or infinite disjunction of the application of the predicate to all elements of the domain.
- Relationship between \(\forall\) and \(\exists\):
  - \(\exists x. P(x)\) is the same as \(\neg \forall x. \neg P(x)\)
  - \(\forall x. P(x)\) is the same as \(\neg \exists x. \neg P(x)\)
Functions

- Consider how to formalize:
  Mary’s father likes music
  One possible way: \( \exists x (f(x, \text{Mary}) \land \text{Likes}(x, \text{Music})) \)
  which means: Mary has at least one father and he likes music.
- We’d like to capture the idea that Mary only has one father.
  - We use functions to capture the single object that can be in relation to another object.
  - Example: \( \text{Likes}(\text{father(Mary)}, \text{Music}) \)
- We can also have \( n \)-ary functions.

Predicate Logic

- syntax (well-formed formulas)
- semantics
- proof theory
  - axiom systems
  - natural deduction
  - sequent calculus
  - resolution principle
The syntax of predicate logic consists of:

- constants
- variables $x, y, \ldots$
- functions $f(), g(), \ldots$
- predicates $P(), Q(), \ldots$
- logical connectives $\land, \lor, \neg, \rightarrow, \leftrightarrow$
- quantifiers $\forall, \exists$
- punctuations: $,,.,(),(,)$

**Definition.** Terms are defined inductively as follows:

- **Base cases**
  - Every constant is a term.
  - Every variable is a term.

- **Inductive cases**
  - If $t_1, t_2, t_3, \ldots, t_n$ are terms then $f(t_1, t_2, t_3, \ldots, t_n)$ is a term, where $f$ is an $n$-ary function.
  - Nothing else is a term.
Predicate Logic

- syntax

**Definition.** Well-formed formulas (wffs) are defined inductively as follows:

- **Base cases:**
  - \( P(t_1,t_2,t_3,\ldots,t_n) \) is a wff, where \( t_i \) is a term, and \( P \) is an \( n \)-ary predicate. These are called atomic formulas.

- **Inductive cases:**
  - If \( A \) and \( B \) are wffs, then so are
    - \( \neg A \), \( A \land B \), \( A \lor B \), \( A \Rightarrow B \), \( A \leftrightarrow B \)
  - If \( A \) is a wff, so is \( \exists x. \ A \)
  - If \( A \) is a wff, so is \( \forall x. \ A \)

- **Nothing else is a wff.**

We often omit the brackets using the same precedence rules as propositional logic for the logical connectives.

Scope and Binding of Variables (I)

- Variables occur both in nodes next to quantifiers and as leaf nodes in the parse tree.
- A variable \( x \) is **bound** if starting at the leaf of \( x \), we walk up the tree and run into a node with a quantifier and \( x \).
- A variable \( x \) is **free** if starting at the leaf of \( x \), we walk up the tree and don’t run into a node with a quantifier and \( x \).

\[
\forall x. (\forall x. (P(x) \land Q(x))) \Rightarrow (\neg P(x) \lor Q(y))
\]
The scope of a variable $x$ is the subtree starting at the node with the variable and its quantifier (where it is bound) minus any subtrees with $\forall x$ or $\exists x$ at their root.

Example:
A wff is closed if it contains no free occurrences of any variable.

$\forall x ((\forall x (P(x) \land Q(x))) \Rightarrow (\neg P(x) \lor Q(y)))$

Scope and Binding of Variables

$\forall x ((P(x) \Rightarrow Q(x)) \land S(x,y))$

Parsing tree:
Scope and Binding of Variables

\[ \forall x ((\exists x (P(x) \Rightarrow Q(x))) \land S(x,y)) \]

Parsing tree:

This is an open formula!

Substitution

Variables are place holders.

- Given a variable \( x \), a term \( t \) and a formula \( P \), we define \( P[t/x] \) to be the formula obtained by replacing each free occurrence of variable \( x \) in \( P \) with \( t \).
- We have to watch out for variable captures in substitution.
Substitution

In order not to mess up with the meaning of the original formula, we have the following restrictions on substitution.

- Given a term \( t \), a variable \( x \) and a formula \( P \),
  
  \[ t \text{ is not free for } x \text{ in } P \]

  if

  - \( x \) in a scope of \( \forall y \) or \( \exists y \) in \( A \); and
  - \( t \) contains a free variable \( y \).

- Substitution \( P[t/x] \) is allows only if \( t \) is free for \( x \) in \( P \).

Substitution

Example:

\[ \forall y (\text{mom}(x) \land \text{dad}(f(y))) \equiv \forall z (\text{mom}(x) \land \text{dad}(f(z))) \]

But

\[ (\forall y (\text{mom}(x) \land \text{dad}(y)))[f(y)/x] = \forall y (\text{mom}(f(y)) \land \text{dad}(f(y))) \not\equiv \]

\[ (\forall z (\text{mom}(x) \land \text{dad}(z)))[f(y)/x] = \forall z (\text{mom}(f(y)) \land \text{dad}(f(z))) \]

[f(y)/x] not allowed since meaning of formulas messed up.
Predicate Logic: Semantics

- Recall that a semantics is a mapping between two worlds.
- A model for predicate logic consists of:
  - a non-empty domain of objects: $D_I$
  - a mapping, called an interpretation that associates the terms of the syntax with objects in a domain
- It’s important that $D_I$ be non-empty, otherwise some tautologies wouldn’t hold such as $(\forall x.A(x)) \Rightarrow (\exists x.A(x))$

Interpretations (Models)

- a fixed element $c' \in D_I$ to each constant $c$ of the syntax
- an $n$-ary function $f': D_I^n \rightarrow D_I$ to each $n$-ary function, $f$, of the syntax
- an $n$-ary relation $R' \subseteq D_I^n$ to each $n$-ary predicate, $R$, of the syntax
Example of a Model

Let’s say our syntax has a constant $c$, a function $f$ (unary), and two predicates $P$, and $Q$ (both binary).

Example: $P(c,f(c))$
In our model, choose the domain to be the natural numbers
- $I(c)$ is 0.
- $I(f)$ is suc, the successor function.
- $I(P)$ is ‘$<$’
- $I(Q)$ is ‘$=$’

Example of an Model

What’s the meaning of $P(c,f(c))$ in this model?

\[
I(P(c,f(c))) = I(c) < I(f(c)) \\
= 0 < suc(I(c)) \\
= 0 < suc(0) \\
= 0 < 1
\]

Which is true.
Valuations

Definition. A valuation \( \nu \), in an interpretation \( I \), is a function from the terms to the domain \( D_I \) such that:

- \( \nu(c) = I(c) \)
- \( \nu(f(t_1, \ldots, t_n)) = f'(\nu(t_1), \ldots, \nu(t_n)) \)
- \( \nu(x) \in D_I \), i.e., each variable is mapped onto some element in \( D_I \)

Example of a Valuation

- \( D_I \) is the set of Natural Numbers
- \( g \) is the function +
- \( h \) is the function \( suc \)
- \( c \) (constant) is 3
- \( y \) (variable) is 1

\[
\nu(g(h(c), y)) = \nu(h(c)) + \nu(y) \\
= suc(\nu(c)) + 1 \\
= suc(3) + 1 \\
= 5
\]
Workout

- $D$ is the set of Natural Numbers
- $g$ is the function $+$
- $h$ is the function $\text{suc}$
- $c$ (constant) is 3
- $y$ (variable) is 1

$\nu(h(h(g(h(y),g(h(y),h(c)))))) = ?$
Workout

Interpret the following formulas with respect to the world (model) in the previous page.

- On(A,Fl) ⇒ Clear(B)
- Clear(B) ∧ Clear(C) ⇒ On(A,Fl)
- Clear(B) ∨ Clear(A)
- Clear(B)
- Clear(C)

Knowledge

Does the following knowledge base (set of formulae) have a model?

- On(A,Fl) ⇒ Clear(B)
- Clear(B) ∧ Clear(C) ⇒ On(A,Fl)
- Clear(B) ∨ Clear(A)
- Clear(B)
- Clear(C)

Figure 15.2

Three Blocks-World Situations
An example

\[ (\forall x) [\text{On}(x, C) \Rightarrow \neg \text{Clear}(C)] \]

Figure 15.2
Three Blocks-World Situations

Closed Formulas

- Recall: A wff is closed if it contains no free occurrences of any variable.
- We will mostly restrict ourselves to closed formulas.
- For formulas with free variables, close the formula by universally quantifying over all its free variables.
Validity (Tautologies)

- **Definition.** A predicate logic formula is **satisfiable** if there is an interpretation and there is a valuation that satisfies the formula (i.e., in which the formula returns T).

- **Definition.** A predicate logic formula is **logically valid (tautology)** if it is true in every interpretation.
  - It must be satisfied by every valuation in every interpretation.

- **Definition.** A wff, $A$, of predicate logic is a **contradiction** if it is false in every interpretation.
  - It must be false in every valuation in every interpretation.

Satisfiability, Tautologies, Contradictions

- A closed predicate logic formula, is **satisfiable** if there is an interpretation $I$ in which the formula returns true.

- A closed predicate logic formula, $A$, is a **tautology** if it is true in every interpretation.
  
  \[ \models A \]

- A closed predicate logic formula is a **contradiction** if it is false in every interpretation.
Tautologies

- How can we check if a formula is a tautology?
- If the domain is finite, then we can try all the possible interpretations (all the possible functions and predicates).
- But if the domain is infinite? Intuitively, this is why a computer cannot be programmed to determine if an arbitrary formula in predicate logic is a tautology (for all tautologies).
- Our only alternative is proof procedures!
- Therefore the soundness and completeness of our proof procedures is very important!

Semantic Entailment

Semantic entailment has the same meaning as it did for propositional logic.

\[ \phi_1, \phi_2, \phi_3 \models \psi \]

means that if \( v(\phi_1) = T \) and \( v(\phi_2) = T \) and \( v(\phi_3) = T \)
then \( v(\psi) = T \), which is equivalent to saying

\[ (\phi_1 \land \phi_2 \land \phi_3) \Rightarrow \psi \]

is a tautology, i.e.,

\[ (\phi_1, \phi_2, \phi_3 \models \psi) \equiv ((\phi_1 \land \phi_2 \land \phi_3) \Rightarrow \psi) \]
An Axiomatic System for Predicate Logic

FO_AL: An extension of the axiomatic system for propositional logic. Use only: \( \Rightarrow \), \( \neg \), \( \forall \)

\[ A \Rightarrow (B \Rightarrow A) \]
\[ (A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C)) \]
\[ (\neg A \Rightarrow \neg B) \Rightarrow (B \Rightarrow A) \]
\[ \forall x. A(x) \Rightarrow A(t), \text{ where } t \text{ is free for } x \text{ in } A \]
\[ \forall x. (A \Rightarrow B) \Rightarrow (A \Rightarrow (\forall x. B)), \text{ where } A \text{ contains no free occurrences of } x \]

FO_AL Rules of Inference

Two rules of inference:

- (modus ponens - MP) From \( A \) and \( A \Rightarrow B \), \( B \) can be derived, where \( A \) and \( B \) are any well-formed formulas.

- (generalization) From \( A \), \( \forall x. A \) can be derived, where \( A \) is any well-formed formula and \( x \) is any variable.
Soundness and Completeness of FO_AL

- FO_AL is sound and complete.
- Completeness was proven by Kurt Gödel in 1929 in his doctoral dissertation.
- Predicate logic is not decidable

Deduction Theorem

- **Theorem.** If \( H \cup \{A\} \vdash B \) by a deduction containing no application of generalization to a variable that occurs free in \( A \), then \( H \vdash_{ph} A \Rightarrow B \)
- **Corollary.** If \( A \) is closed and if \( H \cup \{A\} \vdash B \) then \( H \vdash_{ph} (A \Rightarrow B) \)
### Proof by Refutation

- A **closed** formula is a tautology iff its negation is a contradiction.
- In other words, a closed formula is valid iff its negation is not satisfiable.
- To prove \( \{P_1, \ldots, P_n\} \models S \) is equivalent to prove \( \{P_1, \ldots, P_n, \neg S\} \models \text{false} \).
- This becomes to check if there is an interpretation for \( \{P_1, \ldots, P_n, \neg S\} \).

### Counterexamples

- How can we show a formula is not a tautology?
- Provide a **counterexample**. A counterexample for a closed formula is an interpretation in which the formula does not have the truth value T.
Example

Prove \( \forall x. \forall y. A \vdash \forall y. \forall x. A \)

1. \( \forall x. \forall y. A \) \hspace{1cm} \text{premise}
2. \( \forall x. \forall y. A \Rightarrow \forall y. A \) \hspace{1cm} \text{Ax4}
3. \( \forall y. A \) \hspace{1cm} \text{MP 1, 2}
4. \( \forall y. A \Rightarrow A \) \hspace{1cm} \text{Ax4}
5. \( A \) \hspace{1cm} \text{MP 3, 4}
6. \( \forall x. A \) \hspace{1cm} \text{Gen of 5}
7. \( \forall y. \forall x. A \) \hspace{1cm} \text{Gen of 6}

Workout: Counterexamples

**Show that** \((\forall x. P(x) \lor Q(x)) \Leftrightarrow ((\forall x. P(x)) \lor (\forall x. Q(x)))\)

is not a tautology by constructing a model that makes the formula false.
What does ‘first-order’ mean?

- We can only quantify over variables.
- In higher-order logics, we can quantify over functions, and predicates.
  - For example, in second-order logic, we can express the induction principle:
    \[ \forall P. (P(0) \land (\forall n. P(n) \Rightarrow P(n + 1))) \Rightarrow (\forall n. P(n)) \]
- Propositional logic can also be thought of as zero-order.

A rough timeline in ATP … (1/3)

<table>
<thead>
<tr>
<th>Year</th>
<th>Event</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>450B.C.</td>
<td>Stoics</td>
<td>propositional logic (PL), inference (maybe)</td>
</tr>
<tr>
<td>322B.C.</td>
<td>Aristotle</td>
<td>“syllogisms” (inference rules), quantifiers</td>
</tr>
<tr>
<td>1565</td>
<td>Cardano</td>
<td>probability theory (PL + uncertainty)</td>
</tr>
<tr>
<td>1646</td>
<td>Leibniz</td>
<td>research for a general decision procedure</td>
</tr>
<tr>
<td>-1716</td>
<td></td>
<td>to check the validity of formulas</td>
</tr>
<tr>
<td>1847</td>
<td>Boole</td>
<td>PL (again)</td>
</tr>
<tr>
<td>1879</td>
<td>Frege</td>
<td>first-order logic (FOL)</td>
</tr>
<tr>
<td>1889</td>
<td>Peano</td>
<td>9 axioms for natural numbers</td>
</tr>
</tbody>
</table>
A rough timeline in ATP ...

• To formalize all existing theories to a finite, complete, and consistent set of axioms.
• Decision procedures for all mathematical theories
• 23 open problems.

Resolve the 2nd Hilbert’s problem (in the theory of N)

- Hilbert’s program
- Wittgenstein proof by truth tables
- Gödel completeness theorem of FOL
- Herbrand a proof procedure for FOL based on propositionalization
- Gödel incompleteness theorems for the consistency of Peano axioms
- Gentzen a proof for the consistency of Peano axioms in set theory
- Church, Turing undecidability of FOL
- Gödel a method to prove the consistency of Peano axioms with type theory

Who is to prove the consistency of set theory?
Is type theory consistent?

A rough timeline in ATP ... (3/3)

1954 Davis First machine-generated proof
1955 Beth, Hintikka Semantic Tableaus
1957 Newell, Simon First machine-generated proof in Logic Calculus
1957 Kangar, Prawitz Lazy substitution by free (dummy) Vars
1958 Prawitz First prover for FOL
1959 Gilmore More provers
1960 Davis, Putnam, Longman Davis-Putnam Procedure
1963 Robinson Unification, resolution
Kurt Gödel
1906-1978
• Born an Austro-Hungarian
• 12 years in the Czech
  • refuse to learn Czech
• 23 years in Austria
  • established the completeness of
    1st-order logic in his Ph.D. thesis
• 25 years in Germany
  • 34 years in the U.S.
  • joined Princeton
  • 42 years in America
  • American is in danger of dictatorship
    because I can prove the contradiction
    in American constitution.
  • thought someone was to poison him.
  • ate only his wife’s cooking.
  • 1977, his wife was ill and could not cook.
  • Jan. 1978, died of mal-nutrition.
• 1st Albert Einstein Award, 1951
• Full professor, 1953
• National Science Medal, 1974
• Emeritus professor, 1976
• Einstein, “his work no longer meant much, that he came to the Institute merely… to have the privilege of walking home with Gödel.”

The greatest logician in the 20th century.
One of the greatest achievements in the 20th century.

I knew the general relativity was wrong.

2007/04/03 stopped here.
Predicate Logic: Natural Deduction

Extend the set of rules we used for propositional logic with ones to handle quantifiers.

**Universal Quantification**

\[
\forall x. P \quad \forall \varepsilon
\]
\[
P[t/x]
\]
\[
\forall x. P
\]

\(x_0\) must be arbitrary, meaning it doesn’t appear outside the subproof. \(t\) must be free for \(x\) in \(P\).

**Existential Quantification**

\[
\exists x. P \quad \exists \varepsilon
\]
\[
P[t/x]
\]
\[
\exists x. P
\]

\[
Q
\]

Informally: If we know that the predicate is true for some value, and using an arbitrary variable, we derive that a formula holds, then we can conclude that the formula holds. \(x_0\) must be arbitrary. \(t\) must be free for \(x\) in \(P\).
Example

Show $\forall x. P(x) \Rightarrow Q(x), \forall x. P(x) \vdash \forall x. Q(x)$

1. $\forall x. P(x) \Rightarrow Q(x)$ premise
2. $\forall x. P(x)$ premise
3. $P(x_0) \Rightarrow Q(x_0)$ $\forall e$ 1
4. $P(x_0)$ $\forall e$ 2
5. $Q(x_0)$ $\Rightarrow e$ 4, 5
6. $\forall x. Q(x)$ $\forall i$ 3 – 6

Workout

- Show $P(a), \forall x. P(x) \Rightarrow \neg Q(x) \vdash \neg Q(a)$
- Show $\neg \forall x. P(x) \vdash \exists x. \neg P(x)$
Proof by Refutation

- To prove \( \{P_1, \ldots, P_n\} \models S \) is equivalent to prove that there is no interpretation for \( \{P_1, \ldots, P_n, \neg S\} \).
- But there are infinitely many interpretations!
- Can we limit the range of interpretations?
- Yes, Herbrand interpretations!

Herbrand’s theorem

- Herbrand universe of a formula \( S \)

- Let \( H_0 \) be the set of constants appearing in \( S \).
  - If no constant appears in \( S \), then \( H_0 \) is to consist of a single constant, \( H_0 = \{a\} \).
- For \( i=0,1,2,\ldots \)
  \[ H_{i+1} = H_i \cup \{ f(t_1, \ldots, t_n) | f \text{ is an } n\text{-place function in } S; t_1, \ldots, t_n \in H_i \} \]
- \( H_i \) is called the \( i \)-level constant set of \( S \).
- \( H_\infty \) is the Herbrand universe of \( S \).
Herbrand’s theorem
- Herbrand universe of a formula $S$

Example 1: $S=\{P(a), \neg P(x) \lor P(f(x))\}$
- $H_0=\{a\}$
- $H_1=\{a, f(a)\}$
- $H_2=\{a, f(a), f(f(a))\}$
- $\ldots$
- $H_\infty=\{a, f(a), f(f(a)), f(f(f(a))), \ldots\}$

Example 2: $S=\{P(x) \lor Q(x), R(z), T(y) \lor \neg W(y)\}$
- There is no constant in $S$, so we let $H_0=\{a\}$
- There is no function symbol in $S$, hence $H=H_0=H_1=\ldots=\{a\}$

Example 3: $S=\{P(f(x), a, g(y), b)\}$
- $H_0=\{a, b\}$
- $H_1=\{a, b, f(a), f(b), g(a), g(b)\}$
- $H_2=\{a, b, f(a), f(b), g(a), g(b), f(f(a)), f(f(b)), f(g(a)), f(g(b)), g(f(a)), g(f(b)), g(g(a)), g(g(b))\}$
- $\ldots$
Herbrand’s theorem
- Herbrand universe of a formula $S$

**Expression**
- a term, a set of terms, an atom, a set of atoms, a literal, a clause, or a set of clauses.

**Ground expressions**
- expressions without variables.

It is possible to use a ground term, a ground atom, a ground literal, and a ground clause – this means that no variable occurs in respective expressions.

**Subexpression** of an expression $E$
- an expression that occurs in $E$.

---

Herbrand’s theorem
- Herbrand base of a formula $S$

- **Ground atoms** $P^n(t_1,\ldots,t_n)$
  - $P^n$ is an $n$-place predicate occurring in $S$,
  - $t_1,\ldots,t_n \in H_\infty$

- **Herbrand base of $S$** (atom set)
  - the set of all ground atoms of $S$

- **Ground instance of $S$**
  - obtained by replacing variables in $S$ by members of the Herbrand universe of $S$. 
Herbrand’s theorem

- Herbrand universe & base of a formula S

Example

- \( S = \{P(x), Q(f(y)) \lor R(y)\} \)
- \( C = P(x) \) is a clause in \( S \)
- \( H = \{a, f(a), f(f(a)), \ldots\} \) is the Herbrand universe of \( S \).
- \( P(a), Q(f(a)), Q(a), R(a), R(f(f(a))), \) and \( P(f(f(a))) \) are ground atoms of \( C \).

Workout

\( \{P(x), Q(g(x,y),a) \lor R(f(x))\} \)

- please construct the set of ground terms
- please construct the set of ground atoms
Herbrand’s theorem
- Herbrand interpretation of a formula $S$

- $S$, a set of clauses.
  - i.e., a conjunction of the clauses

- $H$, the Herbrand universe of $S$ and

- $H$-interpretation $\mathcal{I}$ of $S$
  - $\mathcal{I}$ maps all constants in $S$ to themselves.
  - For all $n$-place function symbol $f$ and $h_1,\ldots,h_n$ elements of $H$,
    \[ \mathcal{I}(f(h_1,\ldots,h_n)) = f(h_1,\ldots,h_n) \]

Herbrand’s theorem
- Herbrand interpretation of a formula $S$

- There is no restriction on the assignment to each $n$-place predicate symbol in $S$.
- Let $A=\{A_1, A_2, \ldots, A_n, \ldots\}$ be the atom set of $S$.
- An $H$-interpretation $I$ can be conveniently represented as a subset of $A$.
  - If $A_j \in I$, then $A_j$ is assigned “true”,
  - otherwise $A_j$ is assigned “false”.

---

61

62
Herbrand’s theorem
- *Herbrand interpretation of a formula S*

Example: \( S = \{P(x) \lor Q(x), R(f(y))\} \)

- The Herbrand universe of \( S \) is 
  \[ H = \{a, f(a), f(f(a)), \ldots\}. \]
- Predicate symbols: \( P, Q, R \).
- The atom set of \( S \):
  \[ A = \{P(a), Q(a), R(a), P(f(a)), Q(f(a)), R(f(a)), \ldots\}. \]
- Some \( H \)-interpretations for \( S \):
  - \( I_1 = \{P(a), Q(a), R(a), P(f(a)), Q(f(a)), R(f(a)), \ldots\} \)
  - \( I_2 = \emptyset \)
  - \( I_3 = \{P(a), Q(a), P(f(a)), Q(f(a)), \ldots\} \)

An interpretation of a set \( S \) of clauses does not necessarily have to be defined over the Herbrand universe of \( S \).
Thus an interpretation may not be an \( H \)-interpretation.

Example:
- \( S = \{P(x), Q(y, f(y, a))\} \)
- \( D = \{1, 2\} \)
Herbrand’s theorem
  - Herbrand interpretation of a formula $S$

- But Herbrand is conceptually general enough.

Example (cont.) $S = \{P(x), Q(y, f(y, a))\}$

- $D = \{1, 2\}$
- an interpretation of $S$:

<table>
<thead>
<tr>
<th>$a$</th>
<th>$f(1,1)$</th>
<th>$f(1,2)$</th>
<th>$f(2,1)$</th>
<th>$f(2,2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$P(1)$</th>
<th>$P(2)$</th>
<th>$Q(1,1)$</th>
<th>$Q(1,2)$</th>
<th>$Q(2,1)$</th>
<th>$Q(2,2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$</td>
<td>$F$</td>
<td>$T$</td>
<td>$F$</td>
<td>$T$</td>
<td>$T$</td>
</tr>
</tbody>
</table>

Herbrand’s theorem
  - Herbrand interpretation of a formula $S$

- But Herbrand is conceptually general enough.

Example (cont.) – we can define an $H$-interpretation $I^*$ corresponding to $I$.

- First we find the atom set of $S$
  - $A = \{P(a), Q(a, a), P(f(a, a)), Q(a, f(a, a)), Q(f(a, a), a), Q(f(a, a), f(a, a)), ...\}$

- Next we evaluate each member of $A$ by using the given table
  - $P(a) = P(2) = F$
  - $Q(a, a) = Q(2, 2) = T$
  - $P(f(a, a)) = P(f(2, 2)) = P(1) = T$
  - $Q(a, f(a, a)) = Q(2, f(2, 2)) = Q(2, 1) = F$
  - $Q(f(a, a), a) = Q(f(2, 2), 2) = Q(1, 2) = T$
  - $Q(f(a, a), f(a, a)) = Q(f(2, 2), f(2, 2)) = Q(1, 1) = T$

- Then $I^* = \{Q(a, a), P(f(a, a)), Q(f(a, a), a), ...\}$.
Herbrand’s theorem
- Herbrand interpretation of a formula $S$

- If there is no constant in $S$, the element $a$ used to initiate the Herbrand universe of $S$ can be mapped into any element of the domain $D$.

- If there is more than one element in $D$, then there is more than one $H$-interpretation corresponding to $I$.

Example: $S=\{P(x),Q(y,f(y,z))\}$, $D=\{1,2\}$

<table>
<thead>
<tr>
<th>$f(1,1)$</th>
<th>$f(1,2)$</th>
<th>$f(2,1)$</th>
<th>$f(2,2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$P(1)$</th>
<th>$P(2)$</th>
<th>$Q(1,1)$</th>
<th>$Q(1,2)$</th>
<th>$Q(2,1)$</th>
<th>$Q(2,2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$</td>
<td>$F$</td>
<td>$F$</td>
<td>$T$</td>
<td>$F$</td>
<td>$T$</td>
</tr>
</tbody>
</table>

Two $H$-interpretations corresponding to $I$ are:

- $I=\{Q(a,a),P(f(a,a)),Q(f(a,a),a),\ldots\}$ if $a=2$,
- $I=\{P(a),P(f(a,a)),\ldots\}$ if $a=1$. 


**Herbrand’s theorem**

*Herbrand interpretation of a formula S*

Definition: Given an interpretation \( I \) over a domain \( D \), an \( H \)-interpretation \( I^* \) corresponding to \( I \) is an \( H \)-interpretation that satisfies the condition:

- Let \( h_1, \ldots, h_n \) be elements of \( H \) (the Herbrand universe of \( S \)).
- Let every \( h_i \) be mapped to some \( d_i \) in \( D \).
- If \( P(d_1, \ldots, d_n) \) is assigned \( T \) (\( F \)) by \( I \), then \( P(h_1, \ldots, h_n) \) is also assigned \( T \) (\( F \)) in \( I^* \).

Lemma: If an interpretation \( I \) over some domain \( D \) satisfies a set \( S \) of clauses, then any \( H \)-interpretation \( I^* \) corresponding to \( I \) also satisfies \( S \).

---

**Herbrand’s theorem**

*A set \( S \) of clauses is unsatisfiable if and only if \( S \) is false under all the \( H \)-interpretations of \( S \).*

- We need consider only \( H \)-interpretations for checking whether or not a set of clauses is unsatisfiable.
- Thus, whenever the term “interpretation” is used, a \( H \)-interpretation is meant.
Herbrand’s theorem

Let ∅ denote an empty set. Then:

- A ground instance $C'$ of a clause $C$ is satisfied by an interpretation $I$ if and only if there is a ground literal $L'$ in $C'$ such that $L'$ is also in $I$, i.e. $C' \cap I \neq \emptyset$.
- A clause $C$ is satisfied by an interpretation $I$ if and only if every ground instance of $C$ is satisfied by $I$.
- A clause $C$ is falsified by an interpretation $I$ if and only if there is at least one ground instance $C'$ of $C$ such that $C'$ is not satisfied by $I$.
- A set $S$ of clauses is unsatisfiable if and only if for every interpretation $I$ there is at least one ground instance $C'$ of some clause $C$ in $S$ such that $C'$ is not satisfied by $I$.

Example: Consider the clause $C=\neg P(x) \lor Q(f(x))$. Let $I_1$, $I_2$, and $I_3$ be defined as follows:

- $I_1 = \emptyset$
- $I_2 = \{P(a), Q(a), P(f(a)), Q(f(a)), P(f(f(a))), Q(f(f(a))), \ldots\}$
- $I_3 = \{P(a), P(f(a)), P(f(f(a))), \ldots\}$

$C$ is satisfied by $I_1$ and $I_2$, but falsified by $I_3$.

Example: $S = \{P(x), \neg P(a)\}$.

The only two $H$-interpretations are:

- $I_1 = \{P(a)\}$
- $I_2 = \emptyset$

$S$ is falsified by both $H$-interpretations and therefore is unsatisfiable.
Resolution Principle
- Clausal Forms

Clauses are universally quantified disjunctions of literals; all variables in a clause are universally quantified

\((\forall x_1, \ldots, x_n)(l_1 \lor \cdots \lor l_n)\)

written as

\(l_1 \lor \cdots \lor l_n\)

or

\(\{l_1, \ldots, l_n\}\)

Examples:

\(\{\text{Nat}(s(A)), \neg \text{Nat}(A)\} \quad \{\text{Nat}(s(A)), \neg \text{Nat}(A)\}\)

\(\{\text{Nat}(A)\} \quad \{\text{Nat}(x)\}\)

gives

gives

\(\{\text{Nat}(s(A))\}\)

\(\{\text{Nat}(s(A))\}\)

\(\{\text{Nat}(s(s(A)))\}\)

\(\{\text{Nat}(s(s(x)))\}\)

\(\{\text{Nat}(s(s(s(x))))\}\)


gives

gives

\(\{\text{Nat}(s(s(A)))\}\)


- We need to be able to work with variables!
- Unification of two expressions/literals
Resolution Principle
- Terms and instances

- Consider following atoms

- Ground expressions do not contain any variables

Resolution Principle
- Substitution

A substitution \( s = \{ t_i/v_i, \ldots, t_n/v_n \} \) substitutes variables \( v_i \) for terms \( t_i \) \( (t_i \) does NOT contain \( v_i) \)

Applying a substitution \( s \) to an expression \( \omega \) yields the expression \( \omega s \) which is \( \omega \) with all occurrences of \( v_i \) replaced by \( t_i \)
Workout

Calculate the substitutions for the resolution of the two clauses and the result clauses after the substitutions.

- $\neg P(x), P(f(a)) \lor Q(f(y), g(a, b))$
- $\neg P(g(x, a)), P(y) \lor Q(f(y), g(a, b))$
- $\neg P(g(x, f(a))), P(g(b, y)) \lor Q(f(y), g(a, b))$
- $\neg P(g(f(x), x)), P(g(y, f(f(y)))) \lor Q(f(y), g(a, b)))$

Resolution Principle
- Composing substitutions

Composing substitutions $s_1$ and $s_2$ gives $s_1 \cdot s_2$ which is that substitution obtained by first applying $s_2$ to the terms in $s_1$ and adding remaining term/vars pairs to $s_1$

$$\theta = \{g(x, y)/z\} \cdot \{A/x, B/y, C/w, D/z\} = \{g(A, B)/z, A/x, B/y, C/w\}$$

Apply to

$$P(x, y, z)\theta$$

Gives

$$P(A, B, g(A, B))$$
Resolution Principle
- Properties of substitutions

\[(\omega s_1) s_2 = \omega (s_1 s_2)\]
\[(s_1 s_2) s_3 = s_1 (s_2 s_3)\] associativity
\[s_1 s_2 \neq s_2 s_1\] not commutative

Resolution Principle
- Unification

- Unifying a set of expressions \(\{w_i\}\)
  - Find substitution \(s\) such that \(w_i s = w_j s\) for all \(i, j\)
  - Example
    \[
    \{P(x, f(y), B), P(x, f(B), B)\}
    \]
    \[s = \{B/y, A/x\} \text{ not the simplest unifier}\]
    \[s = \{B/y\} \text{ most general unifier (mgu)}\]

- The most general unifier, the mgu, \(g\) of \(\{w_i\}\) has the property that if \(s\) is any unifier of \(\{w_i\}\) then there exists a substitution \(s'\) such that \(\{w_i\} s = \{w_i\} g s'\)
- The common instance produced is unique up to alphabetic variants (variable renaming)

- *usually we assume there is no common variables in the two atoms*
Workout

P(B,f(x),g(A)) and P(y,z,f(w))
- construct an mgu
- construct a unifier that is not the most general.

Workout

Determine if each of the following sets is unifiable. If yes, construct an mgu.
- \{Q(a), Q(b)\}
- \{Q(a,x), Q(a,a)\}
- \{Q(a,x,f(x)), Q(a,y,y)\}
- \{Q(x,y,z), Q(u,h(v,v),u)\}
- \{P(x_1,g(x_1),x_2,h(x_1,x_2),x_3,k(x_1,x_2,x_3)), P(y_1,y_2,e(y_2),y_3,f(y_2,y_3),y_4)\}
Resolution Principle
- Disagreement set in unification

The disagreement set of a set of expressions \( \{w_i\} \) is the set of subterms \( \{t_i\} \) of \( \{w_i\} \) at the first position in \( \{w_i\} \) for which the \( \{w_i\} \) disagree.

\[
\begin{align*}
\{P(x,A,f(y)), P(w,B,z)\} & \text{ gives } \{x,w\} \\
\{P(x,A,f(y)), P(x,B,z)\} & \text{ gives } \{A,B\} \\
\{P(x,y,f(y)), P(x,B,z)\} & \text{ gives } \{y,B\}
\end{align*}
\]

Resolution Principle
- Unification algorithm

Unify(Terms)
Initialize \( k \leftarrow 0; \)
Initialize \( T_k = Terms; \)
Initialize \( \sigma_k = \{}; \\
* \text{If } T_k \text{ is a singleton, then output } \sigma_k. \text{ Otherwise, continue.} \\
\text{Let } D_k \text{ be the disagreement set of } T_k \text{.} \\
\text{If there exists a var } v_k \text{ and a term } t_k \text{ in } D_k \text{ such that } v_k \text{ does not occur in } t_k, \text{ continue. Otherwise, exit with failure.} \\
\sigma_{k+1} \leftarrow \sigma_k \{t_k / v_k\}; \\
T_{k+1} \leftarrow T_k \{t_k / v_k\}; \\
\hat{k} \leftarrow k + 1; \\
\text{Goto *}
\]
Predicate calculus Resolution

John Allan Robinson (1965)

Let $C_1$ and $C_2$ be two clauses with literals $l_1 \in C_1$ and $\neg l_2 \in C_2$ such that $C_1$ and $C_2$ do not contain common variables, and $\text{mgu}(l_1, l_2) = \theta$

then $C = \left( \{ C_1 - \{ l_1 \} \} \cup \{ C_2 - \{ \neg l_2 \} \} \right) \theta$

is a resolvent of $C_1$ and $C_2$

Given

$C: l_1 \lor l_2 \lor \ldots \lor l_m$
$C: \neg k_1 \lor k_2 \lor \ldots \lor k_n$

$\theta = \text{mgu}(l_1, k_1)$

the resolvent is

$l_2 \theta \lor \ldots \lor l_m \theta \lor k_2 \theta \lor \ldots \lor k_n \theta$

no common variables!
Resolution Principle - Example

\[ P(x) \lor Q(f(x)) \text{ and } R(g(x)) \lor \neg Q(f(A)) \]

Standardizing the variables apart

\[ P(x) \lor Q(f(x)) \text{ and } R(g(y)) \lor \neg Q(f(A)) \]

Substitution \( \theta = \{A/x\} \)

Resolvent \( P(A) \lor R(g(y)) \)

\[ P(x) \lor Q(x,y) \text{ and } \neg P(A) \lor \neg R(B,z) \]

Standardizing the variables apart

Substitution \( \theta = \{A/x\} \)

Resolvent \( Q(A,y) \lor \neg R(B,z) \)

Workout

Find all the possible resolvents (if any) of the following pairs of clauses.

- \( \neg P(x) \lor Q(x,b) \),
Workout

Find all the possible resolvents (if any) of the following pairs of clauses.

- \( \neg P(x) \lor Q(x,b), P(a) \lor Q(a,b) \)
- \( \neg P(x) \lor Q(x,x), \neg Q(a,f(a)) \)
- \( \neg P(x,y,u) \lor \neg P(y,z,v) \lor \neg P(x,v,w) \lor P(u,z,w), P(g(x,y),x,y) \)
- \( \neg P(v,z,v) \lor P(w,z,w), P(w,h(x,x),w) \)

Resolution Principle

- A stronger version of resolution

Use more than one literal per clause

\( \{ P(u), P(v) \} \) and \( \{ \neg P(x), \neg P(y) \} \)

do not resolve to empty clause.

However, ground instances

\( \{ P(A) \} \) and \( \{ \neg P(A) \} \) resolve to empty clause
Resolution Principle
- Factors

Let $C_1$ be a clause such that there exists a substitution $\theta$ that is a mgu of a set of literals in $C_1$. Then $C_1\theta$ is a factor of $C_1$

Each clause is a factor of itself.
Also, \{P(f(y)),R(f(y),y)\} is a factor of \{P(x),P(f(y)),R(x,y)\} with $\theta = \{f(y) / x\}$

Resolution Principle
- Example of refutation

1. \{F(Art,Jon)\} $\Delta$
2. \{F(Bob,Kim)\} $\Delta$
3. \{\neg F(x,y), P(x,y)\} $\Delta$
4. \{\neg P(Art,Jon)\} $\Gamma$

3. \{P(Art,Jon)\} 1, 3
4. \{P(Bob,Kim)\} 2, 3
5. \{\neg F(Art,Jon)\} 3, 4

8. {} 4, 5
9. {} 1, 7
Resolution Principle
- Example

Hypotheses

\(\forall x \ (\text{dog}(x) \Rightarrow \text{animal}(x))\)

\(\text{dog}(\text{fido})\)

\(\forall y \ (\text{animal}(y) \Rightarrow \text{die}(y))\)

Conclusion

\(\text{die}(\text{fido})\)

Clausal Form

\(\neg\text{dog}(x) \lor \text{animal}(x)\)

\(\text{dog}(\text{fido})\)

\(\neg\text{animal}(y) \lor \text{die}(y)\)

Negate the goal

\(\neg\text{die}(\text{fido})\)
Workout (resolution)
- Proof with resolution principle

Hypotheses:
- \( P(m(x), x) \lor Q(m(x)) \)
- \( \neg P(y, z) \lor R(y) \)
- \( \neg Q(m(f(x, y))) \lor \neg T(x, g(y)) \)
- \( S(a) \lor T(f(a), g(x)) \)
- \( \neg R(m(y)) \)
- \( \neg S(x) \lor W(x, f(x, y)) \)

Conclusion
\( W(a, y) \)

Resolution

Properties
- Resolution is sound
- Incomplete

Given \( P(A) \)
Infer \( \{ P(A), P(B) \} \)

But fortunately it is refutation complete
- If KB is unsatisfiable then KB |- \( \square \)
Resolution Principle
- Refutation Completeness

To decide whether a formula \( \text{KB} \models w \), do
- Convert \( \text{KB} \) to clausal form \( \text{KB}' \)
- Convert \( \neg w \) to clausal form \( \neg w' \)
- Combine \( \neg w' \) and \( \text{KB}' \) to give \( \Delta \)
- Iteratively apply resolution to \( \Delta \) and add the results back to \( \Delta \) until either no more resolvents can be added, or until the empty clause is produced.

Resolution Principle
- Converting to clausal form (1/2)

To convert a formula \( \text{KB} \) into clausal form
1. Eliminate implication signs*
   \( (p \Rightarrow q) \) becomes \( \neg p \lor q \)
2. Reduce scope of negation signs*
   \( \neg (p \land q) \) becomes \( \neg p \lor \neg q \)
3. Standardize variables
   \( (\forall x) \neg \exists y (\neg P (x) \lor Q (y)) \) becomes \( (\forall x) \neg P (x) \lor Q (y) \)
4. Eliminate existential quantifiers using **Skolemization**
   * Same as in prop. logic
Resolution Principle
- Converting to clausal form (2/2)

5. Convert to prenex form
   - Move all universal quantifiers to the front
6. Put the matrix in conjunctive normal form*
   - Use distribution rule
7. Eliminate universal quantifiers
8. Eliminate conjunction symbol *
9. Rename variables so that no variable occurs in more than one clause.

Resolution Principle
- Skolemization

Consider \((\forall x) [ (\exists y) \text{Height}(x,y)]\)
The \(y\) depends on the \(x\)

Define this dependence explicitly using a skolem function \(h(x)\)

Formula becomes \((\forall x) [\text{Height}(x,h(x))]\)

General rule is that each occurrence of an existentially quantified variable is replaced by a skolem function whose arguments are those universally quantified variables whose scopes includes the scope of the existentially quantified one

Skolem functions do not yet occur elsewhere !
Resulting formula is not logically equivalent !
Resolution Principle
- Examples of Skolemization

\[ (\forall w) Q(w) \Rightarrow (\exists y) \{ (\forall x)(\exists z) [P(x,y,z) \Rightarrow (\exists u) R(x,y,u,z)] \} \]

gives

\[ (\forall x) [ (\exists y) F(x,y) ] \text{ gives } (\forall x) F(x, h(x)) \]

but

\[ (\exists y) [ (\forall x) F(x,y) ] \text{ gives } [ (\forall x) F(x, s_k) ] \text{ skolem constant} \]

Not logically equivalent!

A well formed formula and its Skolem form are not logically equivalent.
However, a set of formulae is (un)satisfiable if and only if its skolem form is (un)satisfiable.

Resolution Principle
- Example of conversion to clausal form

initial: \( \forall x \ (\forall y \ P(x,y)) \Rightarrow \neg(\forall y \ Q(x,y) \Rightarrow R(x,y)) \)
step 1: \( \forall x \ \neg(\forall y \ P(x,y)) \lor \neg(\forall y \ Q(x,y) \lor R(x,y)) \)
step 2: \( \forall x \ (\exists y \ \neg P(x,y)) \lor (\exists y \ Q(x,y) \land \neg R(x,y)) \)
step 3: \( \forall x \ (\exists y \ \neg P(x,y)) \lor (\exists z \ Q(x,z) \land \neg R(x,z)) \)
step 4: \( \forall x \ \neg P(x,F1(x)) \lor (Q(x,F2(x)) \land \neg R(x,F2(x))) \)
step 5: \( \neg P(x,F1(x)) \lor (Q(x,F2(x)) \land \neg R(x,F2(x))) \)
step 6: \( (\neg P(x,F1(x)) \lor Q(x,F2(x))) \land (\neg P(x,F1(x)) \lor \neg R(x,F2(x))) \)
step 7: \( \{ \neg P(x,F1(x)), Q(x,F2(x)) \} \)
\( \{ \neg P(x,F1(x)), \neg R(x,F2(x)) \} \)
step 8: \( \{ \neg P(x1,F1(x1)), Q(x1,F2(x1)) \} \)
\( \{ \neg P(x2,F1(x2)), \neg R(x2,F2(x2)) \} \)
Workout

Convert the following formula to clausal form.

- $\exists x (P(x) \land \forall y ((\exists z. Q(x,y,s(z))) \rightarrow (Q(x,s(y),x) \land R(y))))$
- $\forall x \forall y (S(x,y,z) \rightarrow \exists z (S(x,z) \land S(z,x)))$

Resolution Principle
- Example of refutation by resolution

Prove $\neg \lor \neg \lor \neg \lor \neg \lor$$

1. $P(x) \lor \neg P(x) \lor \neg I(x,27) \lor \neg I(x,28) \lor S(x,y)$
2. $P(x)$
3. $P(y)$
4. $I(x,27) \lor I(x,28)$
5. $I(x,27)$
6. $I(x,28)$

Prove $I(x,27)$

Figure 16.1
A Resolution Refutation
Resolution Principle

- Search Strategies

- Ordering strategies
  - In what order to perform resolution?
  - Breadth-first, depth-first, iterative deepening?
  - Unit-preference strategy:
    - Prefer those resolution steps in which at least one clause is a unit clause (containing a single literal)

- Refinement strategies
  - Unit resolution: allow only resolution with unit clauses

 Resolution Principle

- Input Resolution

  - at least one of the clauses being resolved is a member of the original set of clauses
  - Input resolution is complete for Horn-clauses but incomplete in general
  - E.g. \{P, Q\}, \{¬P, Q\}, \{P, ¬Q\}, \{¬P, ¬Q\}
  - One of the parents of the empty clause should belong to original set of clauses
Workout

Use input resolution to prove the theorem in page workout(resolution)!

Resolution Principle
- Linear Resolution

- Linear resolvent is one in which at least one of the parents is either
  - an initial clause or
  - the resolvent of the previous resolution step.

- Refutation complete
- Many other resolution strategies exist
workout

Use linear resolution to prove the theorem in page workout(resolution)!

Resolution Principle
- Set of support
  - Ancestor: c2 is a descendant of c1 iff c2 is a resolvent of c1 (and another clause) or if c2 is a resolvent of a descendant of c1 (and another clause); c1 is an ancestor of c2
  - Set of support: the set of clauses coming from the negation of the theorem (to be proven) and their descendants
  - Set of support strategy: require that at least one of the clauses in each resolution step belongs to the set of support
workout

Use set of support to prove the theorem in page workout(resolution)!

Resolution Principle
- Answer extraction

Suppose we wish to prove whether KB |= (∃w)f(w)

We are probably interested in knowing the w for which f(w) holds.

Add Ans(w) literal to each clause coming from the negation of the theorem to be proven; stop resolution process when there is a clause containing only Ans literal
Resolution Principle - Example of answer extraction

1. \( \neg P(x) \lor \neg P(y) \lor \neg I(x,27) \lor \neg I(y,28) \lor S(x,y) \)
   all packages in room 27 are smaller than any of those in 28
2. \( P(A) \)
3. \( P(B) \)
4. \( I(A,27) \lor I(A,28) \)
5. \( I(B,27) \)
6. \( \neg S(B,A) \)
Prove \( (\exists u) I(A,u) \), i.e. in which room is A?

Workout

- Use answer extraction to prove the theorem in page workout(resolution)!
Theory of Equality

- Herbrand Theorem does not apply to FOL with equality.
- So far we’ve looked at predicate logic from the point of view of what is true in all interpretations.
  - This is very open-ended.
- Sometimes we want to assume at least something about our interpretation to enrich the theory in what we can express and prove.
- The meaning of equality is something that is common to all interpretations.
  - Its interpretation is that of equivalence in the domain.
  - If we add \( = \) as a predicate with special meaning in predicate logic, we can also add rules to our various proof procedures.
- Normal models are models in which the symbol \( = \) is interpreted as designating the equality relation.

- An Axiomatic System with Equality

To the previous axioms and rules of inference, we add:

- **EAx1** \( \forall x. x = x \)
- **EAx2** \( \forall x. \forall y. x = y \Rightarrow (A(x, x) \Rightarrow A(x, y)) \)
- **EAx3** \( \forall x. \forall y. x = y \Rightarrow f(x) = f(y) \)
Theory of Equality  
- Natural Deduction Rules for Equality

Reflexivity

\[ t = t \quad = \ i \]

This inference rule is called an **axiom**, because it has no premises.

---

Theory of Equality  
- Natural Deduction Rules for Equality

Substitution

\[
\begin{align*}
\frac{t_1 = t_2 \quad P[t_2/x]}{P[t_1/x]} &= e \\
\frac{t_1 = t_2 \quad P[t_1/x]}{P[t_2/x]} &= e
\end{align*}
\]

where \( t_1 \) and \( t_2 \) are free in \( x \) in \( P \).
Theory of Equality

- Substitution

- Recall: Given a variable \( x \), a term \( t \) and a formula \( P \), we define \( P[t/x] \) to be the formula obtained by replacing **ALL free** occurrence of variable \( x \) in \( P \) with \( t \).

- But with equality, we sometimes don’t want to substitute for all occurrences of a variable.

- When we write \( P[t/x] \) above the line, we get to choose what \( P \) is and therefore can choose the occurrences of a term that we wish to substitute for.

---

Theory of Equality

- Substitution

Recall from existential introduction:

\[
\frac{P[t/x]}{\exists x. P} \quad \text{E} \\
1 \quad \forall x. Q(x, x) \quad \text{premise} \\
2 \quad x_0 \\
3 \quad Q(x_0, x_0) \quad \forall \text{E} 1 \\
4 \quad \exists y. Q(x_0, y) \quad \exists \text{I} 3 \\
5 \quad \forall x. \exists y. Q(x, y) \quad \forall \text{I} 5 \\
\]

- Matching the top of our rule, \( P = Q(x_0, x) \), so line 3 of the proof is \( P[x_0/x] \), which is \( Q(x_0, x) \).

- So we don’t have to substitute in for every occurrence of a term.
Theory of Equality
- Examples

From these two inference rules, we can derive two other properties that we expect equality to have:

- **Symmetry**: \[ \vdash \forall x, y, (x = y) \Rightarrow (y = x) \]
- **Transitivity**: \[ \vdash \forall x, y, z, (x = y) \land (y = z) \Rightarrow (x = z) \]
Theory of Equality
- Example

\[ \forall x, y, z. (x = y) \land (y = z) \Rightarrow (x = z) \]

\[
\begin{align*}
1 & \quad x_0 \\
2 & \quad y_0 \\
3 & \quad (x_0 = y_0) \land (y_0 = z_0) \quad \text{assumption} \\
4 & \quad x_0 = y_0 \quad \land e 3 \\
5 & \quad y_0 = z_0 \quad \land e 3 \\
6 & \quad x_0 = z_0 \quad = e 4, 5 \\
7 & \quad (x_0 = y_0) \land (y_0 = z_0) \Rightarrow (x_0 = z_0) \quad \Rightarrow i 3 - 6 \\
8 & \quad \forall y. (x_0 = y) \land (y = z_0) \Rightarrow (x_0 = z_0) \quad \forall i 2 - 7 \\
9 & \quad \forall x, y. (x = y) \land (y = z) \Rightarrow (x = z) \quad \forall i 1 - 8
\end{align*}
\]

Theory of Equality
- Leibniz’s Law

- The substitution inference rule is related to Leibniz’s Law.

- **Leibniz’s Law:**
  
  if \( t_1 = t_2 \) is a theorem, then so is \( P[t_1 / x] \iff P[t_2 / x] \)

- Leibniz’s Law is generally referred to as the ability to substitute “equals for equals”.
Leibniz

Gottfried Wilhelm von Leibniz (1646-1716)

- The founder of differential and integral calculus.
- Another of Leibniz’s lifelong aims was to collate all human knowledge.

"[He was] one of the last great polymaths - not in the frivolous sense of having a wide general knowledge, but in the deeper sense of one who is a citizen of the whole world of intellectual inquiry."

Theory of Equality
- Example

- From our natural deduction rules, we can derive Leibniz’s Law:
  \[ t_1 = t_2 \vdash P(t_1) \iff P(t_2) \]
Theory of Equality
- Equality: Semantics

- The semantics of the equality symbol is equality on the objects of the domain.
- In ALL interpretations it means the same thing.
- Normal interpretations are interpretations in which the symbol = is interpreted as designating the equality relation on the domain.
- We will restrict ourselves to normal interpretations from now on.

Theory of Equality
- Extensional Equality

- Equality in the domain is extensional, meaning it is equality in meaning rather than form.
- This is in contrast to intensional equality which is equality in form rather than meaning.
- In logic, we are interested in whether two terms represent the same object, not whether they are the same symbols.
- If two terms are intensionally equal then they are also extensionally equal, but not necessarily the other way around.
Theory of Equality

- Equality: Counterexamples

- Show the following argument is not valid:
  \[ \exists x. P(x) \land Q(x), P(A), A = B \models Q(B) \]

- where \( A, B \) are constants

---

Theory of Arithmetic

- Another commonly used theory is that of arithmetic.

- It was formalized by Dedekind in 1879 and also by Peano in 1889.

- It is generally referred to as Peano’s Axioms.

- The model of the system is the natural numbers with the constants 0 and 1, the functions +, *, and the relation <.
Intuitionistic Logic

- "A proof that something exists is constructive if it provides a method for actually constructing it."
- In intuitionistic logic, only constructive proofs are allowed.
- Therefore, they disallow proofs by contradiction. To show \( \phi \), you can’t just show \( \neg \phi \) is impossible.
- They also disallow the law of the excluded middle arguing that you have to actually show one of \( \phi \) or \( \neg \phi \) before you can conclude \( \phi \lor \neg \phi \)
- Intuitionistic logic was invented by Brouwer. Theorem provers that use intuitionistic logic are Nuprl, Coq, Elf, and Lego.
- In this course, we will only be studying classical logic.
Summary

- Predicate Logic (motivation, syntax and terminology, semantics, axiom systems, natural deduction)
- Equality, Arithmetic
- Mechanical theorem proving