4 Predicate / First Order Logic

4.1 Syntax

4.2 Substitutions

4.3 Semantics

4.4 Equivalence and Normal Forms

4.5 Unification

4.6 Proof Procedures

4.7 Implementation of Proof Procedures

4.8 Properties

4.1 Syntax

- **Definition 4.1** An **alphabet** of (first order) predicate logic consists of
  - a finite or countably infinite set $\mathcal{R}$ of relation symbols,
  - a finite or countably infinite set $\mathcal{F}$ of function symbols,
  - a countably infinite set $\mathcal{V}$ of variables,
  - the set $\{\neg/1,\ \wedge/2,\ \vee/2,\ \rightarrow/2,\ \leftrightarrow/2\}$ of connectives,
  - the set $\{\forall,\ \exists\}$ of quantifiers, and
  - the special characters "(", ")", and ",".

Syntax (Continued)

- $\forall$ is called **universal quantifier**, $\exists$ is called **existential quantifier**.
- An **arity** $n \in \mathbb{N}$ is assigned to each function and relation symbol.
- Nullary function symbols are called **constant symbols**.
- **Notation**

$p, q, \ldots$	relation symbols,
p/n	relation symbol with arity n,
$g, h, \ldots$	function symbols,
g/n	function symbol with arity n,
$a, b, \ldots$	constant symbols,
$X, Y, \ldots$	variables,

- **Agreement** In the following we assume that $\mathcal{R}$, $\mathcal{F}$, and $\mathcal{V}$ are given.

Terms

- **Definition 4.2**
  The set $\mathcal{T}(\mathcal{F}, \mathcal{V})$ of terms is the smallest set satisfying the following conditions:

  1. Every variable $X \in \mathcal{V}$ is a term.
  2. If $g/n \in \mathcal{F}$ and $\{t_1, \ldots, t_n\} \subseteq \mathcal{T}(\mathcal{F}, \mathcal{V})$, then the string $g(t_1, \ldots, t_n)$ is a term.

  A term is closed or ground (instantiated), if it does not contain any variables.

  The set of ground terms is denoted by $\mathcal{T}(\mathcal{F})$.

- **Notation** We write g instead of $g()$.

Predicate Logic Formulas

- **Definition 4.3** The set of **atomic formulas** (or, briefly, **atoms**) of the language $\mathcal{L}(\mathcal{R}, \mathcal{F}, \mathcal{V})$ is the set of strings of the form $p(t_1, \ldots, t_n)$, where $p/n \in \mathcal{R}$ and $\{t_1, \ldots, t_n\} \subseteq \mathcal{T}(\mathcal{F}, \mathcal{V})$.

- **Notation** We write p instead of $p()$.

- **Definition 4.4** The set of **predicate logic formulas** of the language $\mathcal{L}(\mathcal{R}, \mathcal{F}, \mathcal{V})$ is the smallest set satisfying the following conditions:

  1. Every atom is a formula.
  2. If F is a formula, then $\neg F$ is a formula.
  3. If F_1 and F_2 are formulas and $\circ/2$ is a binary connective, then $(F_1 \circ F_2)$ is a formula.
  4. If F is a formula, Q is a quantifier and $X \in \mathcal{V}$, then $(QX)\ F$ is a formula.

- **Notation** A (possibly indexed) denotes an atom,
  F, G, H (possibly indexed) denote formulas.

Structural Induction

- Structural induction theorems can be formulated for $\mathcal{T}(\mathcal{F}, \mathcal{V})$ and $\mathcal{L}(\mathcal{R}, \mathcal{F}, \mathcal{V})$.
- Every term in $\mathcal{T}(\mathcal{F}, \mathcal{V})$ has property E, if the following conditions are satisfied:
  1. **Induction basis**: Every variable has property E.
  2. **Induction step**:
     If f/n is an n-ary function symbol, and the terms $t_1, \ldots, t_n$ in $\mathcal{T}(\mathcal{F}, \mathcal{V})$ have property E, then the term $f(t_1, \ldots, t_n)$ also has property E.
- Likewise for formulas $\rightsquigarrow$ **Exercise**.

Structural Recursion

- Structural recursion theorems can be formulated for $\mathcal{T}(\mathcal{F}, \mathcal{V})$ and $\mathcal{L}(\mathcal{R}, \mathcal{F}, \mathcal{V})$.
- There is exactly one function *foo* over the set of terms satisfying the following conditions:
  1. **Recursion base**:
     The value of *foo* for variables and constant symbols is explicitly defined.
  2. **Recursion step**:
     The value of *foo* for $f(t_1, \ldots, t_n)$, $n > 0$, is defined depending only on the values of *foo* for $t_1, \ldots, t_n$.
- **Example**

$$\textit{foo}(T) = \begin{cases} 0 & \text{if } T \text{ is a variable,} \\ 1 & \text{if } T \text{ is a constant symbol,} \\ \sum_{i=1}^{n} \textit{foo}(t_i) & \text{if } T \text{ is of the form } f(t_1, \ldots, t_n). \end{cases}$$

- Likewise for formulas $\rightsquigarrow$ **Exercise**.

Subterms and Subformulas

- **Definition 3.6 (propositional subformulas) can be extended to terms and first order formulas** $\rightsquigarrow$ **Exercise.**

- **Example**

  - $\{g(f(X), Y),\ f(X),\ X,\ Y\}$
    is the set of subterms of the term $g(f(X), Y)$**.**

  - $$\begin{array}{ll}\{ & (\forall X)(\exists Y)\,(q(X) \rightarrow p(g(a,b), f(f(Y)))),\\ & (\exists Y)\,(q(X) \rightarrow p(g(a,b), f(f(Y)))),\\ & (q(X) \rightarrow p(g(a,b), f(f(Y)))),\\ & q(X),\\ & p(g(a,b), f(f(Y))) \qquad\qquad \}\end{array}$$
    is the set of subformulas of the formula
    $(\forall X)(\exists Y)\,(q(X) \rightarrow p(g(a,b), f(f(Y))))$**.**

Free and Bound Occurrences of Variables

- **Definition 4.5** The **free occurrences of a variable** in a predicate logic formula are defined as follows:

  1. The free occurrences of a variable in an atomic formula F are the occurrences of a variable in F.
  2. The free occurrences of a variable in a formula $\neg F$ are the free occurrences of a variable in F.
  3. The free occurrences of a variable in a formula $(F_1 \circ F_2)$ are the free occurrences of a variable in F_1 together with the free occurrences of a variable in F_2.
  4. The free occurrences of a variable in a formula $(QX)\ F$ are the free occurrences of a variable in F without the occurrences of X.

  The occurrence of a variable is called **bound**, if it is not free.

Closed Terms and Formulas

- **Remember** A term is **closed** if it does not contain any occurrence of a variable.
- **Definition 4.6** A **closed** formula (or, briefly, a **sentence**) is a formula, where every occurrence of a variable is bound.

4.2 Substitutions

- **Definition 4.7** A **substitution** is a mapping $\sigma : \mathcal{V} \to \mathcal{T}(\mathcal{F}, \mathcal{V})$, where only a finite number of variables is not mapped to itself.

- Let σ be a substitution:
  - $dom(\sigma) = \{X \mid X \in \mathcal{V} \text{ and } \sigma(X) \neq X\}$.
  - $|dom(\sigma)|$ is finite.
  - σ can be represented by a finite set of pairs

    $$\{X \mapsto \sigma(X) \mid X \in dom(\sigma)\}$$

    and vice versa.
  - If $dom(\sigma) = \emptyset$, then σ is called **empty substitution**.
  - ε denotes an empty substitution.
  - The **restriction** of σ to a set $\mathcal{U} \subseteq \mathcal{V}$ of variables is defined as

    $$\sigma|_{\mathcal{U}} = \{X \mapsto t \mid X \mapsto t \in \sigma \text{ and } X \in \mathcal{U}\}.$$

Instances

- **Notation** Instead of $\sigma(X)$ we will write $X\sigma$ in the sequel.

- **Definition 4.8** Let σ be a substitution. $\sigma : \mathcal{V} \to \mathcal{T}(\mathcal{F}, \mathcal{V})$ is extended to a mapping $\widehat{\sigma} : \mathcal{T}(\mathcal{F}, \mathcal{V}) \to \mathcal{T}(\mathcal{F}, \mathcal{V})$ as follows: Let $t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$, then

$$t\widehat{\sigma} = \begin{cases} t\sigma & \text{if } t \in \mathcal{V}, \\ t & \text{if } t \text{ is a constant,} \\ f(t_1\widehat{\sigma}, \ldots, t_n\widehat{\sigma}) & \text{if } t \text{ is of the form } f(t_1, \ldots, t_n). \end{cases}$$

  $t\widehat{\sigma}$ is called instance of t under $\widehat{\sigma}$.

  If $t\widehat{\sigma}$ is closed, then $t\widehat{\sigma}$ is called ground instance of t under $\widehat{\sigma}$, and $\widehat{\sigma}$ is said to be a ground substitution for t.

Composition of Substitutions

- **Definition 4.9** Let σ and θ be substitutions.
  The **composition** $\sigma\theta$ of σ and θ is the substitution

$$\{X \mapsto t\widehat{\theta} \mid X \mapsto t \in \sigma \text{ and } X \neq t\widehat{\theta}\}$$
$$\cup \{Y \mapsto s \mid Y \mapsto s \in \theta \text{ and } Y \notin \mathit{dom}(\sigma)\}.$$

- **Proposition 4.11** Let σ and θ be substitutions.
  For every term t, we have $t(\widehat{\sigma\theta}) = (t\widehat{\sigma})\widehat{\theta}$.

- **Proof** Structural induction over t $\rightsquigarrow$ **Exercise**.

Composition of Substitutions (continued)

- **Proposition 4.12** Let $t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ and σ, θ and λ be substitutions. Then:
  1. $t(\widehat{(\sigma\theta)\lambda}) = t(\widehat{\sigma(\theta\lambda)})$.
  2. $t(\widehat{\varepsilon\sigma}) = t\widehat{\sigma} = t(\widehat{\sigma\varepsilon})$.

- **Proof**
  1. $$\begin{aligned} & t(\widehat{(\sigma\theta)\lambda}) \\ = \; & (t(\widehat{\sigma\theta}))\widehat{\lambda} && \text{(Proposition 4.11)} \\ = \; & ((t\widehat{\sigma})\widehat{\theta})\widehat{\lambda} && \text{(Proposition 4.11)} \\ = \; & (t\widehat{\sigma})(\widehat{\theta\lambda}) && \text{(Proposition 4.11)} \\ = \; & t(\widehat{\sigma(\theta\lambda)}) && \text{(Proposition 4.11)} \end{aligned}$$
  2. $\rightsquigarrow$ Exercise

  qed

- **Notation** From now on we write σ instead of $\widehat{\sigma}$.

Applications of Substitutions to Formulas

- **Definition 4.13** Let σ be a substitution.

$$\sigma_X = \begin{cases} \sigma & \text{if } X \notin \mathit{dom}(\sigma) \\ \sigma \setminus \{X \mapsto t\} & \text{if } X \mapsto t \in \sigma \end{cases}$$

- **Definition 4.14** Let σ be a substitution.
  The application of a substitution to a formula is defined as follows:

  1. $p(t_1, \ldots, t_n)\sigma = p(t_1\sigma, \ldots, t_n\sigma)$
     for every atom of the form $p(t_1, \ldots, t_n)$.
  2. $(\neg F)\sigma = \neg(F\sigma)$.
  3. $(F_1 \circ F_2)\sigma = (F_1\sigma \circ F_2\sigma)$ for every binary connective $\circ/2$.
  4. $((QX)\ F)\sigma = (QX)\ (F\sigma_X)$ for every quantifier Q.

  If F is a formula and $F\sigma$ is closed,
  then $F\sigma$ is called ground instance of F under σ,
  and σ is said to be a ground substitution for F.

Variants

- **Definition 4.17** Let E_1 and E_2 be either two terms or two predicate logic formulas.

  E_1 and E_2 are called **variants**, if there are substitutions σ and θ such that $E_1 = E_2\sigma$ and $E_2 = E_1\theta$. In this case, we will call E_1 **variant of** E_2 and E_2 **variant of** E_1.

  If E_1 and E_2 are variants and the variables occurring in E_2 have not been used so far in the context, then E_2 is a **new variant** of E_1.

4.3 Semantics

- **What is the meaning of predicate logic formulas?**
- **4.3.1 Interpretations and Models**
- **4.3.2 Herbrand-Interpretations**

Some Preliminaries

- Let D be a set.

$$
\begin{array}{lcll}
D^2 & = & D \times D & = \{(d_1, d_2) \mid d_1 \in D \text{ and } d_2 \in D\} \\
D^n & = & D^{n-1} \times D & = \{(t, d_n) \mid t \in D^{n-1} \text{ and } d_n \in D\} \\
 & & & =: \{(d_1, \ldots, d_n) \mid d_i \in D,\ 1 \leq i \leq n\},\ n > 2 \\
D^1 & = & D & =: \{(d) \mid d \in D\} \\
D^0 & = & & = \{(\,)\}
\end{array}
$$

- **Relations**

$$
\begin{array}{ll}
R \subseteq D^2 & R = \{(n, m) \mid n, m \in \mathbb{N} \text{ and } n < m\} \\
R \subseteq D^3 & R = \{(x, y, z) \mid x, y, z \in \mathbb{N} \text{ and } x^2 + y^2 = z^2\} \\
R \subseteq D^1 & R = \{(n) \mid n \in \mathbb{N} \text{ and } n \text{ even}\} \\
R \subseteq D^0 & R = \emptyset \text{ or } R = \{(\,)\}
\end{array}
$$

Some Preliminaries – Continued

- **Remember** Let D be a set.

$$
\begin{array}{lllll}
D^2 & = & D \times D & = & \{(d_1, d_2) \mid d_1 \in D \text{ and } d_2 \in D\} \\
D^n & = & D^{n-1} \times D & = & \{(t, d_n) \mid t \in D^{n-1} \text{ and } d_n \in D\} \\
 & & & =: & \{(d_1, \ldots, d_n) \mid d_i \in D,\ 1 \leq i \leq n\},\ n > 2 \\
D^1 & = & D & =: & \{(d) \mid d \in D\} \\
D^0 & = & & = & \{(\,)\}
\end{array}
$$

- **Functions**

$$
\begin{array}{lll}
+ : & \mathbb{N}^2 \longrightarrow \mathbb{N} & +(2,3) \longmapsto 5 \\
\mathtt{succ} : & \mathbb{N}^1 \longrightarrow \mathbb{N} & \mathtt{succ}(3) \longmapsto 4 \\
0 : & \mathbb{N}^0 \longrightarrow \mathbb{N} & 0(\,) \longmapsto 0
\end{array}
$$

4.3.1 Interpretations and Models

- **Definition 4.18** A **predicate logic interpretation** I for a predicate logic language $\mathcal{L}(\mathcal{R}, \mathcal{F}, \mathcal{V})$ consists of a non-empty set $\mathcal{D}$ and a mapping $\cdot^I$, which satisfies the following conditions:

  1. Every n-ary function symbol $g/n \in \mathcal{F}$ is mapped to an n-ary function $g^I : \mathcal{D}^n \to \mathcal{D}$.
  2. Every n-ary relation symbol $p/n \in \mathcal{R}$ is mapped to an n-ary relation $p^I \subseteq \mathcal{D}^n$.

  $\mathcal{D}$ is called **domain** of the interpretation.

Variable Assignments

- What is the meaning of a variable?
- **Definition 4.19** A **variable assignment** wrt an interpretation $I = (\mathcal{D}, \cdot^I)$ is a mapping $\mathcal{Z} : \mathcal{V} \longrightarrow \mathcal{D}$.

  The image of a variable X under $\mathcal{Z}$ is denoted by $X^{\mathcal{Z}}$.

  Let $\mathcal{Z}$ be a variable assignment and $d \in \mathcal{D}$.
  $\{X \mapsto d\}\mathcal{Z}$ denotes the variable assignment, where:

$$Y^{\{X \mapsto d\}\mathcal{Z}} = \begin{cases} d & \text{if } Y = X, \\ Y^{\mathcal{Z}} & \text{otherwise.} \end{cases}$$

Interpretation of Terms

- **Definition 4.20** Let $I = (\mathcal{D}, \cdot^I)$ be an interpretation and $\mathcal{Z}$ a variable assignment wrt I. The meaning $t^{I,\mathcal{Z}}$ of a term $t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ is defined as follows:

  1. For every variable $X \in \mathcal{V}$ we have $X^{I,\mathcal{Z}} = X^{\mathcal{Z}}$.
  2. For every term of the form $g(t_1, \ldots, t_n)$ we have

  $$[g(t_1, \ldots, t_n)]^{I,\mathcal{Z}} = g^I(t_1^{I,\mathcal{Z}}, \ldots, t_n^{I,\mathcal{Z}}),$$

  where $g/n \in \mathcal{F}$ and $t_1, \ldots, t_n$ are terms.

Example

- Let $\mathcal{F} = \{h/1,\ g/2,\ a/0\}$.
- $\mathcal{D}_1 = \mathbb{N}$ with successor function $s/1$ and addition $+/2$.
- $\mathcal{D}_2 = \mathbb{Z}$ with predecessor function $p/1$ and subtraction $-/2$.
- $\mathcal{D}_3$ = set of words over $\Sigma = \{\mathsf{o}, \mathsf{m}\}$ with function
  - *add*$_m/1$, which appends m to a word at the right, and function
  - *conc*$/2$, which concatenates two words.

	a	h	g	X^Z	$g(h(a), h(h(a)))$	$g(h(X), g(h(a), h(X)))$
I_1	0	s	$+$	4	3	11
I_2	1	p	$-$	8	1	14
I_3	o	*add*$_m$	*conc*	omo	omomm	omomomomom

Interpretation of Formulas

- **Definition 4.21** Let $I = (\mathcal{D}, \cdot^I)$ be an interpretation and $\mathcal{Z}$ a variable assignment wrt I. I and $\mathcal{Z}$ assign every formula $F \in \mathcal{L}(\mathcal{R}, \mathcal{F}, \mathcal{V})$ a truth value $F^{I,\mathcal{Z}}$ as follows:

  1. $[p(t_1, \ldots, t_n)]^{I,\mathcal{Z}} = \top$ iff $(t_1^{I,\mathcal{Z}}, \ldots, t_n^{I,\mathcal{Z}}) \in p^I$.
  2. $[\neg F]^{I,\mathcal{Z}} = \neg^*(F^{I,\mathcal{Z}})$.
  3. $[(F_1 \circ F_2)]^{I,\mathcal{Z}} = (F_1^{I,\mathcal{Z}} \circ^* F_2^{I,\mathcal{Z}})$ for all binary connectives $\circ/2$.
  4. $[(\forall X)\, F]^{I,\mathcal{Z}} = \top$ iff for all $d \in \mathcal{D}$: $F^{I,\{X \mapsto d\}\mathcal{Z}} = \top$.
  5. $[(\exists X)\, F]^{I,\mathcal{Z}} = \top$ iff for some $d \in \mathcal{D}$: $F^{I,\{X \mapsto d\}\mathcal{Z}} = \top$.

Substitutions and Variable Assignments

- **Lemma 4.21'** Let s, t be terms, G a formula, Y a variable, $I = (\mathcal{D}, \cdot^I)$ an interpretation and $\mathcal{Z}$ a variable assignment wrt I and $d \in \mathcal{D}$. If $[t]^{I,\mathcal{Z}} = d$, then:
  1. $[s\{Y \mapsto t\}]^{I,\mathcal{Z}} = [s]^{I,\{Y \mapsto d\}\mathcal{Z}}$,
  2. $[G\{Y \mapsto t\}]^{I,\mathcal{Z}} = [G]^{I,\{Y \mapsto d\}\mathcal{Z}}$ if t is free for Y in G.
- **Proof** induction on the structure of s and G $\rightsquigarrow$ **Exercise**.

Models

- **Proposition 4.21"**
  If $F \in \mathcal{L}(\mathcal{R}, \mathcal{F}, \mathcal{V})$ is closed, then $F^{I,\mathcal{Z}} = F^{I,\mathcal{Z}'}$ for any I, $\mathcal{Z}$ and $\mathcal{Z}'$.
- **Proof** $\rightsquigarrow$ **Exercise**.
  - If formulas are closed, then their meaning is independent of a variable assignment.
- **Notation** If F is closed, then we write F^I instead of $F^{I,\mathcal{Z}}$.
- **Definition 4.22**
  Let $I = (\mathcal{D}, \cdot^I)$ be an interpretation and $F \in \mathcal{L}(\mathcal{R}, \mathcal{F}, \mathcal{V})$ be closed.
  I is called **model** for F, in symbols $I \models F$, if $F^I = \top$ holds.

Some Remarks

- In the sequel we consider only closed formulas if not mentioned otherwise.
- Many notions and results known from propositional logic can be extended to predicate logic. For example:
  - Validity, satisfiability, falsifiability, unsatisfiability.
  - E.g., a closed formula F is valid iff all interpretations are models for F.
  - Theorem 3.12 (extended)
    A closed formula F is valid iff $\neg F$ is unsatisfiable.
  - Theorem 3.15 (extended) Let F, F_1, $\ldots$, F_n be closed formulas.
    $\{F_1, \ldots, F_n\} \models F$ iff $\models (\langle F_1, \ldots, F_n \rangle \rightarrow F)$.

Logical Consequence

- **Definition 4.23** A closed formula F is a (predicate logic) consequence of a set $\mathcal{G}$ of closed formulas, in symbols $\mathcal{G} \models F$,
  iff every model for $\mathcal{G}$ is also a model for F.

Propositional vs. First Order Logic

- What is the relation between propositional and first order logic?
- If all relation symbols in $\mathcal{R}$ are nullary,
  then first order logic is equivalent to propositional logic.
- If no variables are occurring in the formulas,
  then first order logic is equivalent to propositional logic.

4.3.2 Herbrand-Interpretations

- In the following we assume that $\mathcal{F}$ contains at least one constant. Otherwise we add a symbol $a/0$ to $\mathcal{F}$.

- **Definition 4.24** An interpretation $I = (\mathcal{D}, \cdot^I)$ for a predicate logic language $\mathcal{L}(\mathcal{R}, \mathcal{F}, \mathcal{V})$ is a **Herbrand-interpretation**, if the following conditions are satisfied:

  1. $\mathcal{D} = \mathcal{T}(\mathcal{F})$; it is called **Herbrand-universe**.
  2. For every $t \in \mathcal{T}(\mathcal{F})$ we have $t^I = t$.

- $\mathcal{F} = \{a/0,\ b/0,\ h/1,\ g/2\}$.
  $\mathcal{T}(\mathcal{F}) = \{a,\ b,\ h(a),\ h(b),\ g(a,a),\ g(a,b),\ g(b,a),\ g(b,b),\ h(h(a)),\ \ldots\}$.

Herbrand-Interpretations and Formulas

- $\mathcal{L}(\{p/1, q/1, r/1\}, \{g/1, a/0\}, \mathcal{V})$
- $F = ((\forall X)\,(p(X) \vee q(X)) \wedge (\forall Y)\, r(g(Y)))$
- $\mathcal{T}(\mathcal{F}) = \{a,\ g(a),\ g(g(a)),\ \ldots\}$

I_1: $p^{I_1} = q^{I_1} = r^{I_1} = \mathcal{T}(\mathcal{F})$. $\rightsquigarrow$ I_1 is a model for F.

I_2: $p^{I_2} = q^{I_2} = r^{I_2} = \emptyset$. $\rightsquigarrow$ I_2 is not a model for F.

I_3: $p^{I_3} = \{g(g(a)),\ g(g(g(g(a)))),\ \ldots\}$, $q^{I_3} = \{a,\ g(a)\}$, $r^{I_3} = \{g(g(a))\}$.
$\rightsquigarrow$ I_3 is not a model for F.

- **Notation**
  - I_1 = $\{p(a),\ q(a),\ r(a),\ p(g(a)),\ q(g(a)),\ r(g(a)),\ \ldots\}$
  - I_2 = $\emptyset$
  - I_3 = $\{p(g(g(a))),\ p(g(g(g(g(a))))),\ \ldots,\ q(a),\ q(g(a)),\ r(g(g(a)))\}$

4.4 Equivalence and Normal Forms

4.4.1 Semantic Equivalence

4.4.2 Prenex Normal Form

4.4.3 Skolem Normal Form

4.4.4 Clause Form

4.4.1 Semantic Equivalence

- In this subsection, formulas need not to be closed.
- Two formulas F and G are called semantically equivalent, in symbols $F \equiv G$ if $F^{I,\mathcal{Z}}$ = $G^{I,\mathcal{Z}}$ holds for all interpretations I and all variable assignments $\mathcal{Z}$ wrt I.
- Note For closed formulas we find $G \equiv F$ if $G^I = F^I$ for all interpretations I.
- Theorem 3.17 (propositional semantic equivalences) holds also for first-order formulas.

Some Semantic Equivalences

- **Theorem 4.25** Let F and G be formulas.
  The following equivalences hold:

$$
\begin{array}{rcll}
\neg(\forall X)\, F & \equiv & (\exists X)\, \neg F & \\
\neg(\exists X)\, F & \equiv & (\forall X)\, \neg F & \\
((\forall X)\, F \wedge (\forall X)\, G) & \equiv & (\forall X)\, (F \wedge G) & \\
((\exists X)\, F \vee (\exists X)\, G) & \equiv & (\exists X)\, (F \vee G) & \\
(\forall X)\, (\forall Y)\, F & \equiv & (\forall Y)\, (\forall X)\, F & \\
(\exists X)\, (\exists Y)\, F & \equiv & (\exists Y)\, (\exists X)\, F & \\
((\forall X)\, F \wedge G) & \equiv & (\forall X)\, (F \wedge G), & \text{if } X \text{ does not occur free in } G. \\
((\forall X)\, F \vee G) & \equiv & (\forall X)\, (F \vee G), & \text{if } X \text{ does not occur free in } G. \\
((\exists X)\, F \wedge G) & \equiv & (\exists X)\, (F \wedge G), & \text{if } X \text{ does not occur free in } G. \\
((\exists X)\, F \vee G) & \equiv & (\exists X)\, (F \vee G), & \text{if } X \text{ does not occur free in } G.
\end{array}
$$

Proof for $\neg(\forall X)\, F \equiv (\exists X)\, \neg F$

- Let $I = (\mathcal{D}, \cdot^I)$ be an interpretation and $\mathcal{Z}$ a variable assignment wrt I. We find:

$$
\begin{aligned}
[\neg(\forall X)\, F]^{I,\mathcal{Z}} = \top \quad & \textbf{iff} \quad \neg^*[(\forall X)\, F]^{I,\mathcal{Z}} = \top \\
& \textbf{iff} \quad [(\forall X)\, F]^{I,\mathcal{Z}} = \bot \\
& \textbf{iff} \quad F^{I,\{X \mapsto d\}\mathcal{Z}} = \top \text{ does not hold for all } d \in D \\
& \textbf{iff} \quad F^{I,\{X \mapsto d\}\mathcal{Z}} = \bot \text{ holds for some } d \in D \\
& \textbf{iff} \quad \neg^* F^{I,\{X \mapsto d\}\mathcal{Z}} = \top \text{ holds for some } d \in D \\
& \textbf{iff} \quad [(\exists X)\, \neg F]^{I,\mathcal{Z}} = \top.
\end{aligned}
$$

qed

Standardizing Apart

- Replacement Theorem 3.18 holds also for predicate logic formulas.
- **Definition 4.26** The variables occurring in a formula F are **standardized apart**, if no two quantifiers occurring in F bind the same variable, and no variable occurs both free and bound.
- **Proposition 4.27** For every formula, there is a semantically equivalent formula where the variables are standardized apart.
- **Auxiliary Proposition** Let Q be a quantifier, F a formula of the form $(QX)\,G$, and Y a variable not occurring in F. Then $F \equiv (QY)\,G\{X \mapsto Y\}$ holds.
- **Proof** $\rightsquigarrow$ **Exercise**.

Proof Sketch of Proposition 4.27

- **Proof**
  - Let F be a predicate logic formula.
  - We find only finitely many, say n, occurrences of subformulas of the form $(QX)G$ in F.
  - We prove by induction on n that each of these occurrences can be replaced according to the auxiliary proposition.
  - Let G be the formula obtained from F in this way.
  - The variables occurring in G are standardized apart. qed
- **Agreement** From now on we assume wlog that variables are standardized apart.

4.4.2 Prenex Normal Form

- **Definition 4.28** A formula F is in prenex normal form, if it is of the form $(Q_1 X_1) \ldots (Q_n X_n)\, G$, where $Q_i \in \{\forall, \exists\}$ and $n \geq 0$ holds, $X_1, \ldots, X_n$ are variables, and G itself does not contain any further quantifier.

  G is called matrix of F.

- **Proposition 4.29** There is an algorithm which transforms each sentence F into a semantically equivalent sentence F' in prenex normal form.

An Algorithm for Transformation into Prenex Normal form

- While F is not in prenex normal form apply one of the following rules:

$$\frac{\neg(\forall X)\, G}{(\exists X)\, \neg G} \qquad \frac{\neg(\exists X)\, G}{(\forall X)\, \neg G}$$

$$\frac{((QX)\, G \wedge H)}{(QX)\, (G \wedge H)} \qquad \frac{(G \wedge (QX)\, H)}{(QX)\, (G \wedge H)} \qquad \frac{((QX)\, G \vee H)}{(QX)\, (G \vee H)} \qquad \frac{(G \vee (QX)\, H)}{(QX)\, (G \vee H)}$$

- **Example**

$$\begin{aligned}
& (\neg(\exists X)\, (\forall Y)\, p(X,Y) \wedge (\forall Z)\, q(Z)) \\
\equiv\ & ((\forall X)\, \neg(\forall Y)\, p(X,Y) \wedge (\forall Z)\, q(Z)) \\
\equiv\ & ((\forall X)\, (\exists Y)\, \neg p(X,Y) \wedge (\forall Z)\, q(Z)) \\
\equiv\ & (\forall X)\, ((\exists Y)\, \neg p(X,Y) \wedge (\forall Z)\, q(Z)) \\
\equiv\ & (\forall X)\, (\exists Y)\, (\neg p(X,Y) \wedge (\forall Z)\, q(Z)) \\
\equiv\ & (\forall X)\, (\exists Y)\, (\forall Z)(\neg p(X,Y) \wedge q(Z)).
\end{aligned}$$

- **Soundness and termination** $\rightsquigarrow$ **Exercise.**

4.4.3 Skolem Normal Form

- **Idea** We eliminate all existential quantifiers.

- **Definition 4.30** Let $\mathcal{L}(\mathcal{R}, \mathcal{F}, \mathcal{V})$ be a predicate logic language.

  Let $\mathcal{F}_S$ be a countable set of function symbols such that $\mathcal{F}_S \cap \mathcal{F} = \emptyset$ and $\mathcal{F}_S$ contains countably many function symbols for each arity. The elements of $\mathcal{F}_S$ are called **Skolem function symbols**. Nullary Skolem function symbols are also called **Skolem constant symbols**.

- We now consider $\mathcal{L}(\mathcal{R}, \mathcal{F} \cup \mathcal{F}_S, \mathcal{V})$.

- **Definition 4.31** A formula is in **Skolem normal form** if it is of the form $(\forall X_1) \ldots (\forall X_n)\, G$, where $n \geq 0$ holds, $X_1, \ldots, X_n$ are variables, and G itself does not contain any further quantifier.

Transformation into Skolem Normal Form

- Let F be a formula in prenex normal form (variables are standardized apart).
- While F is not in Skolem normal form apply the following rule:

$$\frac{(\forall X_1)\ \ldots\ (\forall X_n)\,(\exists Y)\,G}{(\forall X_1)\ \ldots\ (\forall X_n)\,(G\{Y \mapsto g(X_1,\ldots,X_n)\})}$$

- **Theorem 4.32** If G is a Skolem normal form of the sentence F, then F is satisfiable iff G is satisfiable.
- The transformation into Skolem normal form preserves satisfiability.
- Alternatively we could have eliminated the universal quantifiers:
  - dual Skolem Normal Form,
  - validity preserving.

Proof Sketch of Theorem 4.32

- Suppose F is in prenex normal form and variables are standardized apart.
- **Auxiliary Proposition** Let F be a formula in prenex normal form, where the variables are standardized apart, and F' be the formula obtained from F by a single application of the replacement rule. Then the following holds: F is satisfiable iff F' is satisfiable.

  Proof of Auxiliary Proposition $\rightsquigarrow$ **Exercise.**
- Let E be the proposition: F' is a formula in prenex normal form, where all variables are standardized apart, and is satisfiable iff F is satisfiable.
- With $F = F'$ proposition E holds before entering the while-loop.
- From the Auxiliary Proposition we learn that E is a loop invariant.
- From Theorem 3.25 we learn that E holds after the loop has been left.
- The loop is left only if F' is in Skolem normal form. qed

4.4.4 Clause Form

- Let F be a sentence and H a Skolem normal form of F.
- F is satisfiable iff H is satisfiable.
- H is of the form $\forall G = (\forall X_1) \ldots (\forall X_n)\, G$,
  where $X_1, \ldots, X_n$ are all the variables occurring in F.
- There are no quantifiers occurring in the matrix G.
- Every variable occurring in H is universally quantified.
- We can transform G into clause form.
- Let G' be a formula in clause form which is semantically equivalent to G.
  - F is satisfiable iff $\forall G'$ is satisfiable.
  - F is unsatisfiable iff $\forall G'$ is unsatisfiable.

Example: Proving Mathematical Theorems by Machines

- **Theorem: Let G be a group with $\cdot$ as binary operator in infix notation and e as identity element. If, for all elements X of G, the property $X \cdot X = e$ holds, then G is a commutative group.**

- **A group G satisfies the following axioms:**

  A_1 **If $X, Y \in G$, then also $X \cdot Y \in G$.**

  A_2 **For all $X, Y, Z \in G$ holds $(X \cdot (Y \cdot Z)) = ((X \cdot Y) \cdot Z)$.**

  A_3 **For all $X \in G$ holds $X \cdot e = e \cdot X = X$.**

  A_4 **For every $X \in G$ there exists a unique (inverse) element $X^{-1} \in G$ with property $X \cdot X^{-1} = X^{-1} \cdot X = e$.**

- **We represent $X \cdot Y = Z$ by $p(X, Y, Z)$ and X^{-1} by $f(X)$.**

Proving Mathematical Theorems by Machines (continued)

- **The group axioms can be represented as follows:**

A_1' $\quad (\forall X_1)(\forall Y_1)(\exists Z_1)\, p(X_1, Y_1, Z_1)$

A_2' $\quad ((\forall X_2)(\forall Y_2)(\forall Z_2)(\forall U_2)(\forall V_2)(\forall W_2)$
$\langle (p(X_2, Y_2, U_2),\ p(Y_2, Z_2, V_2),\ p(U_2, Z_2, W_2)\rangle \rightarrow p(X_2, V_2, W_2))$
$\wedge$
$(\forall X_3)(\forall Y_3)(\forall Z_3)(\forall U_3)(\forall V_3)(\forall W_3)$
$(\langle p(X_3, Y_3, U_3),\ p(Y_3, Z_3, V_3),\ p(X_3, V_3, W_3)\rangle \rightarrow p(U_3, Z_3, W_3)))$

A_3' $\quad ((\forall X_4)\, p(X_4, e, X_4) \wedge (\forall X_5)\, p(e, X_5, X_5))$

A_4' $\quad ((\forall X_6)\, p(X_6, f(X_6), e) \wedge (\forall X_7)\, p(f(X_7), X_7, e))$

Proving Mathematical Theorems by Machines (continued)

- The conclusion of the theorem can be represented as follows:

$$C' \quad ((\forall X_8)\, p(X_8, X_8, e) \\ \rightarrow \; ((\forall U_9)(\forall V_9)(\forall W_9)\, (p(U_9, V_9, W_9) \rightarrow p(V_9, U_9, W_9))))$$

- Hence, the theorem is represented by the formula:

$$F = (\langle A'_1,\ A'_2,\ A'_3,\ A'_4\rangle \rightarrow C')$$

- The theorem is proven if F is valid, or, equivalently, if

$$\neg F = \langle A'_1,\ A'_2,\ A'_3,\ A'_4,\ \neg C'\rangle$$

is unsatisfiable.

Proving Mathematical Theorems by Machines (continued)

- **As clause form of $\neg F$ we obtain:**

$$
\begin{array}{ll}
\forall \langle & [p(X_1, Y_1, g(X_1, Y_1))], \\
& [\neg p(X_2, Y_2, U_2),\ \neg p(Y_2, Z_2, V_2),\ \neg p(U_2, Z_2, W_2),\ p(X_2, V_2, W_2)], \\
& [\neg p(X_3, Y_3, U_3),\ \neg p(Y_3, Z_3, V_3),\ \neg p(X_3, V_3, W_3),\ p(U_3, Z_3, W_3)], \\
& [p(X_4, e, X_4)], \\
& [p(e, X_5, X_5)], \\
& [p(X_6, f(X_6), e], \\
& [p(f(X_7), X_7, e)], \\
& [p(X_8, X_8, e)], \\
& [p(a, b, c)], \\
& [\neg p(b, a, c)] \qquad \rangle.
\end{array}
$$

- **How can we show unsatisfiablity of a predicate logic formula?**

4.5 Unification

- We consider terms from $\mathcal{T}(\mathcal{F}, \mathcal{V})$.
- **Definition 4.33** An **equation** is an expression of the form $s \approx t$, where s and t are terms.
- **Definition 4.34** A **unification problem** consists of a multiset of equations

$$\dot{\{}s_1 \approx t_1, \ldots, s_n \approx t_n\dot{\}}$$

and is the question, whether there exists a substitution σ such that $s_i\sigma = t_i\sigma$ holds for all $1 \leq i \leq n$.

If such a substitution σ exists, then the terms s_i and t_i, $1 \leq i \leq n$, are said to be **simultaneously unifiable** and σ is said to be a **unifier** for the unification problem.

Example

- $\dot{\{}X \approx a,\ Y \approx Z,\ g(W, X) \approx g(W, a)\dot{\}}$
- **Unifiers:**
  - $\sigma_1 = \{X \mapsto a,\ Y \mapsto Z\}$
  - $\sigma_2 = \{X \mapsto a,\ Z \mapsto Y\}$
  - $\sigma_3 = \{X \mapsto a,\ Y \mapsto a,\ Z \mapsto a\}$
  - $\sigma_4 = \{X \mapsto a,\ Y \mapsto g(a, U),\ Z \mapsto g(a, U),\ W \mapsto a\}$
- **Do you prefer unifiers over other unifiers?**

Most General Unifiers

- **Definition 4.35** Let σ and θ be substitutions. σ is more general than θ, in symbols $\sigma \geq \theta$, if there is a substitution λ such that $\sigma\lambda = \theta$ holds.
- **Definition 4.36** Two substitutions σ and θ are said to be variants, in symbols $\sigma \sim \theta$, if $\sigma \geq \theta$ and $\theta \geq \sigma$ hold.
- **Definition 4.37** Let U be a unification problem.
  A substitution σ is a most general unifier for U, if σ is a unifier for U and $\sigma \geq \theta$ holds for every unifier θ for U.
- **Theorem 4.38** (Unification Theorem) Let $\mathcal{U}$ be a solvable unification problem.
  There exists a most general unifier for $\mathcal{U}$.
- **Proof**
  - (i) Specification of a unification algorithm.
  - (ii) Proof of Termination $\rightsquigarrow$ see lecture Foundations of LP.
  - (iii) Proof of Soundness. $\rightsquigarrow$ see lecture Foundations of LP.

Unification Algorithm

Input A unification problem $\mathcal{U}$.
Output A most general unifier θ for $\mathcal{U}$, if $\mathcal{U}$ is solvable,
or **not unifiable**, otherwise.

$\theta := \varepsilon$.
While $\mathcal{U}$ is non-empty do:
Select an equation $s \approx t$ from $\mathcal{U}$.
$\mathcal{U} := \mathcal{U} \dot{\setminus} \{s \approx t\}$.
Apply one of the following rules:

(1) If $s \approx t$ is of the form $X \approx r$ or $r \approx X$ and variable X does not occur in the term r, then $\theta := \theta\{X \mapsto r\}$ and $\mathcal{U} := \mathcal{U}\{X \mapsto r\}$.
(2) If $s \approx t$ is of the form $X \approx X$, then do nothing.
(3) If $s \approx t$ is of the form $X \approx r$ or $r \approx X$, variable X occurs in the term r and $r \neq X$ holds, then terminate with **not unifiable**.
(4) If $s \approx t$ is of the form $f(s_1, \ldots, s_n) \approx f(t_1, \ldots, t_n)$, then add the equations $s_1 \approx t_1, \ldots, s_n \approx t_n$ to $\mathcal{U}$.
(5) If $s \approx t$ is of the form $f(s_1, \ldots, s_n) \approx g(t_1, \ldots, t_m)$ where $f \neq g$, then terminate with **not unifiable**.

4.6 Proof Methods

- Resolution
- Semantic Tableaus
- Calculus of Natural Deduction
- Hilbert Systems
- Sequent Calculus
- Connection Method

4.6.1 Resolution

- It is a negative and analyzing calculus.
  - The alphabet is the alphabet of predicate logic.
  - The language is the set of predicate logic formulas in clause form.
  - The only axiom is the empty clause.
  - The derivation rules are the resolution and the factorization rule.

The Resolution Rule

- **Definition 4.42** Let

$$C_1 = [p(s_1, \ldots, s_k), L_1, \ldots, L_m]$$

and

$$C_2 = [\neg p(t_1, \ldots, t_k), L_{m+1}, \ldots, L_n]$$

be clauses, where $k, m, n \geq 0$.

If $\dot{\{} s_i \approx t_i \mid 1 \leq i \leq k \dot{\}}$ is unifiable with mgu σ then

$$C = [L_1, \ldots, L_n]\sigma$$

is called resolvent of C_1 and C_2 wrt $p(s_1, \ldots, s_k)$ and $\neg p(t_1, \ldots, t_k)$.

C has been obtained by applying the resolution rule to C_1 and C_2, where $p(s_1, \ldots, s_k)$ and $\neg p(t_1, \ldots, t_k)$ are said to be the literals resolved upon.

The Factoring Rule

- **Definition 4.43** Let

$$C = [p(s_1, \ldots, s_k), p(t_1, \ldots, t_k), L_1, \ldots, L_m]$$

or

$$C = [\neg p(s_1, \ldots, s_k), \neg p(t_1, \ldots, t_k), L_1, \ldots, L_m]$$

be clauses, where $k, m \geq 0$.

If $\dot{\{} s_i \approx t_i \mid 1 \leq i \leq k \dot{\}}$ is unifiable with mgu σ then

$$C' = [p(t_1, \ldots, t_k), L_1, \ldots, L_m]\sigma$$

and

$$C' = [\neg p(t_1, \ldots, t_k), L_1, \ldots, L_m]\sigma,$$

are factors of C, respectively.

C' has been obtained by applying the factoring rule to C.

Resolution Derivations and Refutations

- **Definition 4.44** Let $F = \forall \langle C_1, \ldots, C_n \rangle$ be a sentence in clause form, where C_i, $1 \leq i \leq n$, are clauses.

  1. The sequence $(C_i \mid 1 \leq i \leq n)$ is a **resolution derivation** for C.
  2. If $(C_i \mid 1 \leq i \leq m)$ is a resolution derivation for F, and C_{m+1} is obtained by applying the resolution or the factoring rule to new variants of elements from $(C_i \mid 1 \leq i \leq m)$, then $(C_i \mid 1 \leq i \leq m+1)$ is a **resolution derivation** for F.
  3. A resolution derivation for F which contains the empty clause $[\,]$ is called **resolution refutation** for F.

- **Note**

  - A resolvent can be computed from two variants of one clause.
  - F may contain already $[\,]$.
  - It suffices to consider refutations in which $[\,]$ occurs only once.
  - We may assume that $[\,]$ is the last clause in a refutation.

Example

- **Is the following formula unsatisfiable?**

$$\neg((p(0) \wedge (\forall X)(p(X) \rightarrow p(h(X)))) \rightarrow p(h(h(h(h(h(0))))))$$

- **Clause form:**

$$(\forall X)\ \langle [p(0)],\ [\neg p(X),\ p(h(X))],\ [\neg p(h(h(h(h(h(0)))))]\rangle.$$

- **Resolution refutation of the matrix:**

1	$[p(0)]$	
2	$[\neg p(X),\ p(h(X))]$	
3	$[\neg p(h(h(h(h(h(0)))))]$	
4	$[\neg p(X_1),\ p(h(h(X_1)))]$	**res(2,2)**
5	$[\neg p(X_3),\ p(h(h(h(h(X_3)))))]$	**res(4,4)**
6	$[p(h(h(h(h(0)))))]$	**res(1,5)**
7	$[\,]$	**res(3,6)**

Another Example

- "The barber shaves all male inhabitants, who do not shave themselves."

$$(\forall X)(\neg p(X, X) \leftrightarrow p(b, X))$$

- Clause form:

$$(\forall X)(\forall Y)\ \langle [p(X, X),\ p(b, X)],\ [\neg p(b, Y),\ \neg p(Y, Y)] \rangle.$$

- Resolution refutation of the matrix:

1	$[p(X, X),\ p(b, X)]$	
2	$[\neg p(b, Y),\ \neg p(Y, Y)]$	
3	$[p(b, b)]$	fac(1)
4	$[\neg p(b, b)]$	fac(2)
5	$[\,]$	res(3,4)

Resolution Proof

- **Definition 4.43"** Let $F = \forall \langle C_1, \ldots, C_n \rangle$ be a predicate logic formula in clause form and $S = (C_i \mid 1 \leq i \leq m)$ a resolution derivation for F. The length of S is $m - n$.
- **Definition 4.44** Let F be a sentence and G a clause form of $\neg F$.
  A resolution proof for F is a resolution refutation for G.

  F is called theorem of the resolution calculus,
  if there exists a resolution proof for F.

  We denote with $\vdash_r F$ that there exists a resolution proof for F.

Proving Mathematical Theorems by Machines (continued)

1	$[p(X_1, Y_1, g(X_1, Y_2))]$	
2	$[\neg p(X_2, Y_2, U_2), \neg p(Y_2, Z_2, V_2), \neg p(U_2, Z_2, W_2), p(X_2, V_2, W_2)]$	
3	$[\neg p(X_3, Y_3, U_3), \neg p(Y_3, Z_3, V_3), \neg p(X_3, V_3, W_3), p(U_3, Z_3, W_3)]$	
4	$[p(X_4, e, X_4)]$	
5	$[p(e, X_5, X_5)]$	
6	$[p(X_6, f(X_6), e)]$	
7	$[p(f(X_7), X_7, e)]$	
8	$[p(X_8, X_8, e)]$	
9	$[p(a, b, c)]$	
10	$[\neg p(b, a, c)]$	
11	$[p(X_9, V_2, e), \neg p(Y_9, U_9, V_9), \neg p(X_2, Y_2, U_2)]$	res$(2, 8)$
13	$[p(X_{10}, V_{10}, W_{10}), \neg p(e, Z_{10}, W_{10}), \neg p(X_{10}, Z_{10}, V_{10})]$	res$(2, 8)$
14	$[p(X_{11}, V_{11}, U_{11}), \neg p(Y_{11}, e, V_{11}), \neg p(X_{11}, Y_{11}, U_{11})]$	res$(2, 4)$
40	$[p(X_{12}, e, W_{12}), \neg p(e, X_{12}, W_{12})]$	res$(13, 8)$
43	$[p(X_{13}, V_{13}, Z_{13}), \neg p(X_{13}, Z_{13}, V_{13})]$	res$(13, 5)$
67	$[p(X_{14}, g(X_{14}, Z_{14}), Z_{14})]$	res$(43, 1)$
3239	$[p(e, V_{15}, Y_{15}), \neg p(Y_{15}, e, V_{15})]$	res$(14, 5)$
3325	$[\neg p(Y_{16}, e, a), \neg p(b, Y_{16}, c)]$	res$(10, 14)$
3627	$[\neg p(g(b, c), e, a)]$	res$(3325, 67)$
4630	$[p(e, Z_{17}, X_{17}), \neg p(X_{17}, Z_{17}, e)]$	res$(3239, 43)$
5199	$[p(Z_{18}, e, X_{18}), \neg p(X_{18}, Z_{18}, e)]$	res$(40, 4630)$
5800	$[\neg p(a, g(b, c), e)]$	res$(3627, 5199)$
518609	$[p(a, V_{19}, e), \neg p(b, c, V_{19})]$	res$(11, 9)$
645778	$[p(a, g(b, c), e)]$	res$(518609, 1)$
646029	$[\,]$	res$(645778, 5800)$

4.8 Properties

4.8.1 Herbrand interpretations

- For each interpretation there is a corresponding Herbrand interpretation.

4.8.2 Soundness and completeness theorem

- $\models F$ iff $\vdash_r F$.

4.8.3 Compactness theorem

- $\mathcal{F}$ is satisfiable iff each finite subset of $\mathcal{F}$ is satisfiable.

4.8.4 Undecidability theorem

- The question whether $\mathcal{F} \models G$ is undecidable in first-order logic.

Completeness of the Resolution Method – Overview

$\{F_1, \ldots, F_n\} \models F$

$\Updownarrow$ Theorem 3.15 (extended)

$(\langle F_1, \ldots, F_n\rangle \rightarrow F)$ is valid

$\Updownarrow$ Theorem 3.12 (extended)

$\neg(\langle F_1, \ldots, F_n\rangle \rightarrow F)$ is unsatisfiable

$\Downarrow$ Theorem 4.32

G is unsatisfiable, where G is a clause form of $\neg(\langle F_1, \ldots, F_n\rangle \rightarrow F)$

$\Updownarrow$ Theorem 4.55

$\{C\sigma \mid C$ is clause in G and σ is ground substitution for $C\}$ is unsatisfiable

$\Updownarrow$ Bijection ground atoms – propositional variables

H is unsatisfiable, where H is the set of propositional clauses corresponding to $\{C\sigma \mid \ldots\}$

$\Downarrow$ Corollary 3.40

finite subset $H' \subseteq H$ is unsatisfiable

$\Downarrow$ Theorem 3.42

there is a propositional resolution refutation for H'

$\Updownarrow$ Bijection ground atoms – propositional variables

there is a first-order resolution refutation for H'

$\Downarrow$ Lemma 4.58

there is a first-order resolution refutation for G

$\Updownarrow$ Definition 4.44

there is a first-order resolution proof for $(\langle F_1, \ldots, F_n\rangle \rightarrow F)$

4.8.1 Herbrand Interpretations

- $\mathcal{F} = \{a/0,\ f/1\}$ and $\mathcal{R} = \{p/1,\ q/1,\ r/1\}$.
- $G = (\forall X)(\forall Y)\langle[p(X),\ q(X)],\ [r(f(Y))]\rangle$
- Herbrand universe: $\mathcal{T}(\mathcal{F}) = \{a,\ f(a),\ f(f(a))\ \ldots\}$.
- Herbrand interpretations, e.g.:
  - $I_1 = \{p(a),\ q(a),\ r(f(a)),\ p(f(a)),\ q(f(a)),\ r(f(f(a))),\ \ldots\}$
  - $I_2 = \emptyset$.
- Different interpretations, e.g.: $I_3 = (\mathcal{D}, \cdot^{I_3})$ where
  - $\mathcal{D} = \{0, 1\}$,
  - $a \mapsto 0$,
  - $f/1 \mapsto f^*/1 : \mathcal{D} \longrightarrow \mathcal{D}$ with $f^*(0) = 1$ and $f^*(1) = 0$,
  - $p^I = \{0\},\ q^I = \{0\},\ r^I = \{0\}$.
- We can construct a corresponding Herbrand interpretation!

Corresponding Herbrand Interpretations

- **Definition 4.51** Let $I = (\mathcal{D}, \cdot^I)$ be an interpretation and $\{t_1, \ldots, t_n\} \subseteq \mathcal{T}(\mathcal{F})$. A Herbrand interpretation J is called corresponding to I if the following condition is satisfied: for all relation symbols p/n holds:

$$p(t_1, \ldots, t_n)^I = \top \text{ iff } p(t_1, \ldots, t_n) \in J.$$

Example

- Consider again $I_3 = (\mathcal{D}, \cdot^{I_3})$ with
  - $\mathcal{D} = \{0, 1\}$,
  - $a \mapsto 0$,
  - $f/1 \mapsto f^*/1 : \mathcal{D} \longrightarrow \mathcal{D}$ mit $f^*(0) = 1$ and $f^*(1) = 0$,
  - $p^I = \{0\},\ q^I = \{0\},\ r^I = \{0\}$.
- Set of ground atoms:

$$\{p(a),\ q(a),\ r(a),\ r(f(a)),\ p(f(a)),\ q(f(a)),\ r(f(f(a))),\ \ldots\}$$

- Meaning under I_3:

$p(a)$	$q(a)$	$r(a)$	$r(f(a))$	$p(f(a))$	$q(f(a))$	$r(f(f(a)))$	$\ldots$
$\top$	$\top$	$\top$	$\bot$	$\bot$	$\bot$	$\top$	$\ldots$

- Corresponding Herbrand interpretation:

$$J_3 = \{p(a),\ q(a),\ r(a),\ p(f(f(a))),\ \ldots\}$$

Interpretation vs. Herbrand Interpretation

- **Lemma 4.52** If an interpretation I is a model for a predicate logic sentence F in Skolem normal form, then every Herbrand interpretation corresponding to I is also a model for F.
- **Proof** $\rightsquigarrow$ **Exercise**.
- **Implication** The real numbers cannot be characterized within predicate logic.
  - Suppose, the real numbers can be characterized.
  - Then there exists a sentence F such that for all $I = (D, \cdot^I)$ we find: $F^I = \top$ **iff** D is uncountable.
  - Select I with $F^I = \top$.
  - Let J be the Herbrand interpretation corresponding to I.
  - Then $F^J = \top$.
  - But the domain of Herbrand interpretations is countable.
  - **Contradiction** qed

Interpretations vs. Herbrand Interpretations (continued)

- **Theorem 4.53** A predicate logic sentence F in Skolem normal form is unsatisfiable iff F is mapped to $\perp$ under each Herbrand interpretation.
- **Proof**

$\Rightarrow$ immediate.

$\Leftarrow$ Suppose, F is mapped to $\perp$ under each Herbrand interpretation, but is satisfiable.

  - Then there exists an I with $F^I = \top$.
  - Let J be the Herbrand interpretation corresponding to I.
  - By Lemma 4.52 we find $F^J = \top$.
  - Contradiction qed

Interpretations vs. Herbrand Interpretations (cont.)

- Lemma 4.52 and Theorem 4.53 hold only for formulas in Skolem normal form.
- **Example** Let $\mathcal{F} = \{a/0\}$, $\mathcal{R} = \{p/1\}$ and $X \in \mathcal{V}$.
  Consider $F = (p(a) \wedge (\exists X)\neg p(X))$.
  - Let $I = (\{1,2\}, \cdot^I)$ with $a^I = 1$ and $p^I = \{1\}$. Then,

$$\begin{aligned}
(p(a) \wedge (\exists X)\neg p(X))^I &= p(a)^I \wedge^* ((\exists X)\neg p(X))^I \\
&= a^I \in p^I \wedge^* ((\exists X)\neg p(X))^I \\
&= 1 \in \{1\} \wedge^* ((\exists X)\neg p(X))^I \\
&= \top \wedge^* ((\exists X)\neg p(X))^I \\
&= ((\exists X)\neg p(X))^I \\
&= \top \text{ because } 2 \notin \{1\}
\end{aligned}$$

  - $\mathcal{T}(\mathcal{F}) = \{a\}$
  - $I' = \{p(a)\}$ is the Herbrand interpretation corresponding to I.
  - $F^{I'} = \bot \rightsquigarrow$ Lemma 4.52 does not carry over.
  - For $I'' = \emptyset$ we find $F^{I''} = \bot \rightsquigarrow$ Theorem 4.53 does not carry over.

Further Properties

- **Proposition 4.54** Let F be a predicate logic formula, I a Herbrand interpretation, $\mathcal{Z}$ a variable assignment wrt I, and $t_1, \ldots, t_n \in \mathcal{T}(\mathcal{F})$. Then,

$$F^{I,\{X_1 \mapsto t_1\} \ldots \{X_n \mapsto t_n\}\mathcal{Z}} = [F\{X_1 \mapsto t_1, \ldots, X_n \mapsto t_n\}]^{I,\mathcal{Z}}.$$

- **Proof** $\rightsquigarrow$ **Exercise**.

Further Properties (continued)

- **Theorem 4.55** Let $F = \forall \langle C_1, \ldots, C_n \rangle$ be a sentence in clause form. F is satisfiable iff $\{C_i\sigma \mid 1 \leq i \leq n, \sigma$ is a ground substitution for $C_i\}$ is satisfiable.
- **Proof** Let $X_1, \ldots, X_k$ be all variables occurring in $\langle C_1, \ldots, C_n \rangle$.
  - F is satisfiable iff there exists a Herbrand model for F (Lemma 4.52).
  - $I \models F$ iff $[\langle C_1, \ldots, C_n \rangle]^{I, \{X_1 \mapsto t_1\} \ldots \{X_k \mapsto t_k\} \mathcal{Z}} = \top$
    for all $t_1, \ldots, t_k \in \mathcal{T}(\mathcal{F})$ and arbitrary $\mathcal{Z}$
    iff $[\langle C_1, \ldots, C_n \rangle \{X_1 \mapsto t_1, \ldots, X_k \mapsto t_k\}]^I = \top$
    for all $t_1, \ldots, t_k \in \mathcal{T}(\mathcal{F})$ (Proposition 4.54)
    iff $[\langle C_1, \ldots, C_n \rangle \sigma]^I = \top$
    for all ground substitutions σ for $\langle C_1, \ldots, C_n \rangle$
    iff $[C_i\sigma]^I = \top$
    for $1 \leq i \leq n$ and all ground substitutions σ for C_i
    iff $I \models \{C_i\sigma \mid 1 \leq i \leq n,\ \sigma$ ground substitution for $C_i\}$.

qed

Example

- Let $\mathcal{F} = \{a/0,\ s/1\}$, $\mathcal{R} = \{p/1,\ q/1\}$ and $X, Y \in \mathcal{V}$.
- Consider:

$$\begin{aligned} F &= \forall\, \langle C_1, C_2 \rangle \\ &= (\forall X)(\forall Y)\ \langle [p(X)], [\neg q(s(Y))] \rangle. \end{aligned}$$

- We obtain:

$$\begin{aligned} G &= \{C_i\sigma \mid 1 \leq i \leq 2, \sigma \text{ ground substitution for } C_i\} \\ &= \{[p(a)],\ [p(s(a))],\ \ldots\} \cup \{[\neg q(s(a))],\ [\neg q(s(s(a)))],\ \ldots\} \end{aligned}$$

- Obviously:

$$I \models F \text{ iff } I \models G.$$

4.8.2 Soundness and Completeness Theorems

- **First Order Resolution Lemma 4.56**
  Let $F = \forall\langle C_1, \ldots, C_n\rangle$ be a first order sentence in clause form with clauses C_i, $1 \leq i \leq n$, and let C be a resolvent or a factor of new variants of clauses from F. Then $F \equiv \forall\,\langle C_1, \ldots, C_n, C\rangle$.
- **Proof Observe**: $\forall\,\langle C_1, \ldots, C_n, C\rangle \equiv (F \wedge \forall\, C)$.
  To show: for all I: $I \models F$ iff $I \models (F \wedge \forall\, C)$.

$\Leftarrow$ If $I \models (F \wedge \forall\, C)$ then $I \models F$ for all I.

$\Rightarrow$ Suppose, $I \models F$.

1st C is a factor $\rightsquigarrow$ **Exercise**.

2nd Let $C = [L_1, \ldots, L_n]\sigma$ resolvent of

$$\begin{array}{lcl} C' & = & [p(s_1, \ldots, s_k), L_1, \ldots, L_m] \text{ and} \\ C'' & = & [\neg p(t_1, \ldots, t_k), L_{m+1}, \ldots, L_n], \end{array}$$

where C' and C'' are new variants of clauses from F and σ is a mgu for $\dot{\{}s_i \approx t_i \mid 1 \leq i \leq k\dot{\}}$.

Proof of Lemma 4.56 (continued)

- Suppose, I is not a model for $\forall C$.
- Then we find a variable assignment $\mathcal{Z}$ wrt. I with

$$[L_1\sigma]^{I,\mathcal{Z}} \vee^* \ldots \vee^* [L_n\sigma]^{I,\mathcal{Z}} = \bot. \tag{1}$$

- Because $I \models F$ we find: $I \models \forall\,(C'\sigma)$ and $I \models \forall\,(C''\sigma)$.
- Hence:

$$[p(s_1,\ldots,s_k)\sigma]^{I,\mathcal{Z}} \vee^* [L_1\sigma]^{I,\mathcal{Z}} \vee^* \ldots \vee^* [L_m\sigma]^{I,\mathcal{Z}} = \top \tag{2}$$

and

$$[\neg p(t_1,\ldots,t_k)\sigma]^{I,\mathcal{Z}} \vee^* [L_{m+1}\sigma]^{I,\mathcal{Z}} \vee^* \ldots \vee^* [L_n\sigma]^{I,\mathcal{Z}} = \top \tag{3}$$

- Because of (1) and (2) we conclude: $[p(s_1,\ldots,s_k)\sigma]^{I,\mathcal{Z}} = \top$.
- Because of (1) and (3) we conclude: $[\neg p(t_1,\ldots,t_k)\sigma]^{I,\mathcal{Z}} = \top$.
- Contradiction, because $s_i\sigma = t_i\sigma$ for all $1 \leq i \leq k$. qed

Example

- **Consider the sentence:**

$$((\forall X)p(X) \wedge (\forall Y)\,(p(Y) \rightarrow q(Y))).$$

- **It's clause form is:**

$$(\forall X)(\forall Y)\,\langle[p(X)], [\neg p(Y), q(Y)]\rangle.$$

- **As a resolvent of new variants of the two clauses we obtain:**

$$[q(Z)].$$

- **We find:**

$$\forall\,\langle[p(X)], [\neg p(Y), q(Y)]\rangle \equiv \forall\,\langle[p(X)], [\neg p(Y), q(Y)], [q(Z)]\rangle.$$

Generalized Resolution Lemma

- **Lemma 4.56'** Let $F = \forall \langle C_1, \ldots, C_n \rangle$ be a predicate logic sentence in clause-form and $(C_i \mid 1 \leq i \leq m)$ a resolution derivation for F. Then:

$$F \equiv \forall \langle C_1, \ldots, C_m \rangle.$$

- **Proof**
  Structural induction on the length $k = m - n$ of the resolution derivation.

  IB $k = 0$: immediate because $n = m$.

  IH Suppose the proposition holds for k.

  IS Let $(C_i \mid 1 \leq i \leq m + 1)$ be a resolution derivation of length $k + 1$.

  - From IH we conclude: $F \equiv \forall \langle C_1, \ldots, C_m \rangle$.
  - From Lemma 4.56 we conclude:
    $\forall \langle C_1, \ldots, C_m \rangle \equiv \forall \langle C_1, \ldots, C_m, C_{m+1} \rangle$.
  - Because $\equiv$ is transitive, we conclude $F \equiv \forall \langle C_1, \ldots, C_{m+1} \rangle$.

  ▷ The proposition follows by an application of the induction theorem.

qed

Lifting-Lemma 1

- **Lemma 4.57** (**Lifting-Lemma 1**) Let C_1 and C_2 be clauses and σ a ground substitution for C_1 and C_2.

  (i) If C is a resolvent of $C_1\sigma$ and $C_2\sigma$, then there exists a resolvent C' of C_1 and C_2 as well as a substitution λ, such that $C'\lambda = C$ holds.

  (ii) If C is a factor of $C_1\sigma$, then there exists a factor C' of C_1 as well as a substitution λ, such that $C'\lambda = C$ holds.

Proof of Lifting-Lemma 1

- **Proof**

  (i) $C_1 = [p(s_1, \ldots, s_k), L_1, \ldots, L_m], C_2 = [\neg p(t_1, \ldots, t_k), L_{m+1}, \ldots, L_n]$.

  - $s_i\sigma = t_i\sigma$ for $1 \leq i \leq k$,
    because σ is a ground substitution for C_1 and C_2.
  - $C = [L_1, \ldots, L_n]\sigma$.
  - σ is a unifier for $\mathcal{U} = \dot{\{}s_1 \approx t_1, \ldots, s_k \approx t_k\dot{\}}$.
  - We find an mgu θ for $\mathcal{U}$ and λ, such that: $\theta\lambda = \sigma$ (Theorem 4.38).
  - $C' = [L_1, \ldots, L_n]\theta$ is resolvent for C_1 and C_2.
  - $C'\lambda = [L_1, \ldots, L_n]\theta\lambda = [L_1, \ldots, L_n]\sigma = C$.

- **(ii) Exercise** qed

Example

- **Consider:**

$$C_1 = [\neg p(W)],\ C_2 = [p(f(X)), \neg q(X)],\ \sigma = \{X \mapsto a,\ W \mapsto f(a)\}.$$

- **Then:**

$$C_1\sigma = [\neg p(f(a))],\ C_2\sigma = [p(f(a)), \neg q(a)].$$

- **As resolvent of $C_1\sigma$ and $C_2\sigma$ we obtain:**

$$C = [\neg q(a)]$$

- **As resolvent of C_1 and C_2 with mgu $\theta = \{W \mapsto f(X)\}$ we obtain**

$$C' = [\neg q(X)]$$

- **With $\lambda = \{X \mapsto a\}$ we find $C'\lambda = C$ and $\theta\lambda = \sigma$.**

Lifting Lemma 2

- **Lemma 4.58 (Lifting Lemma 2)**
  Let C_j be clauses and σ_j ground substitutions for C_j, $1 \leq j \leq n$. If there is a resolution derivation $(B_i \mid 1 \leq i \leq m)$ for $\langle C_1\sigma_1, \ldots, C_n\sigma_n\rangle$, then there is a resolution derivation $(B'_i \mid 1 \leq i \leq m)$ for $\forall \langle C_1, \ldots, C_n\rangle$ and a sequence $(\lambda_i \mid 1 \leq i \leq m)$ of substitutions such that $B'_i\lambda_i = B_i$ for all $1 \leq i \leq m$.
- **Proof** Induction on the length $l = m - n$ of the derivation $(B_i \mid 1 \leq i \leq m)$.

  IB $l = 0$: In this case $n = m$ and the proposition follows with $B'_i = C_i$ and $\lambda_i = \sigma_i$ for all $1 \leq i \leq n$.

  IH Suppose the proposition holds for l.

  IS Let $(B_i \mid 1 \leq i \leq m+1)$ be a resolution derivation of length $l+1$. Because of IH we find a resolution derivation $(B'_i \mid 1 \leq i \leq m)$ of length l for $\forall \langle C_1, \ldots, C_n\rangle$ and a sequence $(\lambda_i \mid 1 \leq i \leq m)$ of substitutions with $B'_i\lambda_i = B_i$ for all $1 \leq i \leq m$. $(*)$

  1st B_{m+1} is a factor $\rightsquigarrow$ Exercise

  2nd B_{m+1} is a resolvent $\rightsquigarrow$ next slide

Proof Lifting Lemma 2 (Cont.)

- Let B_{m+1} be the resolvent of B_j and B_k with $j, k \leq m$.
- Because of $(*)$ we find B'_j and λ_j with $B'_j\lambda_j = B_j$.
- Let B''_j be a new variant of B'_j. Then there is a substitution θ_j with $B''_j\theta_j = B'_j$.
- Alltogether: $B_j = B'_j\lambda_j = (B''_j\theta_j)\lambda_j = B''_j(\theta_j\lambda_j)$.
- Analogously we find B''_k, λ_k and θ_k with $B_k = B'_k\lambda_k = (B''_k\theta_k)\lambda_k = B''_k(\theta_k\lambda_k)$.
- Because B''_j and B''_k are new variants, we conclude:
  dom$((\theta_j\lambda_j)|_{Var(B''_j)}) \cap$ *dom*$((\theta_k\lambda_k)|_{Var(B''_k)}) = \emptyset$.
- Let $\sigma = (\theta_j\lambda_j)|_{Var(B''_j)} \cup (\theta_k\lambda_k)|_{Var(B''_k)}$.
- Hence we obtain: $B''_j\sigma = B_j$ and $B''_k\sigma = B_k$.
- Because of Lifting Lemma 1 we find a resolvent B'_{m+1} of B''_j and B''_k as well as a substitution λ_{m+1} with $B'_{m+1}\lambda_{m+1} = B_{m+1}$.
- Together with $(*)$ we obtain a resolution derivation $(B_i \mid 1 \leq i \leq m+1)$ and a sequence $(\lambda_i \mid 1 \leq i \leq m+1)$ with $B'_i\lambda_i = B_i$ for all $1 \leq i \leq m+1$.
- An application of the induction theorem yields the desired result. qed

Example 1

- **Consider the clauses**

$$[\neg p(W)],\ [p(f(X)), \neg q(X)],\ [q(g(Y)), \neg r(Y)],\ [r(Z)]$$

with

$$\sigma = \{W \mapsto f(g(a)),\ X \mapsto g(a),\ Y \mapsto a,\ Z \mapsto a\}.$$

- **We obtain the following derivations:**

B_1	$[\neg p(f(g(a)))]$	B'_1	$[\neg p(W)]$	
B_2	$[p(f(g(a))), \neg q(g(a))]$	B'_2	$[p(f(X)), \neg q(X)]$	
B_3	$[q(g(a)), \neg r(a)]$	B'_3	$[q(g(Y)), \neg r(Y)]$	
B_4	$[r(a)]$	B'_4	$[r(Z)]$	
B_5	$[\neg q(g(a))]$	B'_5	$[\neg q(X_2)]$	**res(1,2)**
B_6	$[\neg r(a)]$	B'_6	$[\neg r(Y_3)]$	**res(3,5)**
B_7	$[\,]$	B'_7	$[\,]$	**res(4,6)**

Example 2

- **Consider the clauses**

$$[p(a)],\ [\neg p(X), p(f(X))],\ [\neg p(Y), p(f(Y))],\ [\neg p(f(f(a)))]$$

with

$$\sigma = \{X \mapsto a,\ Y \mapsto f(a)\}.$$

- **We obtain the derivations:**

B_1	$[p(a)]$	B'_1	$[p(a)]$	
B_2	$[\neg p(a), p(f(a))]$	B'_2	$[\neg p(X), p(f(X))]$	
B_3	$[\neg p(f(a)), p(f(f(a)))]$	B'_3	$[\neg p(Y), p(f(Y))]$	
B_4	$[\neg p(f(f(a)))]$	B'_4	$[\neg p(f(f(a)))]$	
B_5	$[\neg p(a), p(f(f(a)))]$	B'_5	$[\neg p(X_1), p(f(f(X_1)))]$	**res(2,3)**
B_6	$[p(f(f(a)))]$	B'_6	$[p(f(f(a)))]$	**res(1,5)**
B_7	$[\,]$	B'_7	$[\,]$	**res(4,6)**

Closed Formulas and Propostional Logic

- Let F be a generalized conjunction of ground-instantiated clauses.

$$F = \langle [p(a)],\ [\neg p(f(f(a)))],\ [\neg p(a),\ p(f(a))],\ [\neg p(f(a)),\ p(f(f(a)))] \rangle$$

- Let G be the set of ground atoms occurring in F.

$$G = \{p(a),\ p(f(a)),\ p(f(f(a)))\}$$

- Let V be a sufficiently large set of propositional variables.

$$V = \{p_1,\ p_2,\ p_3\}$$

- We find a bijection between G and V.

$$p(a) \Leftrightarrow p_1,\ p(f(a)) \Leftrightarrow p_2,\ p(f(f(a))) \Leftrightarrow p_3$$

- All propositional results are applicable to F.

The Resolution Theorem

- **Theorem 4.59 (Resolution Theorem)**
  Let $F = \forall \langle C_1, \ldots, C_l \rangle$ be a sentence in clause form.
  F is unsatisfiable iff there is a resolution refutation for F.
- **Proof**

$\Leftarrow$ Let $(C_i \mid 1 \leq i \leq m)$ be a resolution refutation for F.

- Lemma 4.56': $F \equiv \forall \langle C_1, \ldots, C_m \rangle$.
- Because of $C_m = [\,]$ and Theorem 3.17 (unsatisfiability) we conclude $F \equiv [\,]$.
- Hence, F is unsatisfiable.

Proof of the Resolution Theorem (cont.)

$\Rightarrow$ **Suppose, F is unsatisfiable.**

- Then, $\mathcal{G} = \{C_i\sigma \mid 1 \leq i \leq l,\ \sigma$ ground substitution for $C_i\}$ is unsatisfiable (Theorem 4.55).
- To each ground atom occurring in $\mathcal{G}$ we can assign a propositional variable such that any two atoms are mapped to different variables.
- Then, we find a finite subset $\{C_1\sigma_1, \ldots, C_m\sigma_m\} \subseteq \mathcal{G}$, which is propositionally unsatisfiable (Corollary 3.40).
- Then, we find a propositional resolution refutation for $\langle C_1\sigma_1, \ldots, C_m\sigma_m\rangle$ (Theorems 3.12 and 3.42).
- Then, we find a predicate logic resolution refutation for $\forall\, \langle C_1, \ldots, C_m\rangle$ (Lemma 4.58).
- By construction each C_j, $1 \leq j \leq m$, is a clause occurring in F.
- Hence, we obtain a resolution refutation for F (by eliminating variants of $\{C_1, \ldots, C_m\}$). qed

Example

- **Consider:**

$$F = \langle [p(a)],\ [\neg p(X),\ p(f(X))],\ [\neg p(f(f(a)))] \rangle.$$

- **The set of all ground instances of clauses occurring in F:**

$$\{[p(a)],\ [\neg p(f(f(a)))],\ [\neg p(a),\ p(f(a))],\ [\neg p(f(a)),\ p(f(f(a)))], \ldots\}.$$

- **Ground atoms can be replaced by propositional variables:**

$$\{[p_1],\ [\neg p_3], [\neg p_1,\ p_2],\ [\neg p_2, p_3], \ldots\}.$$

- **We find a finite subset which is propositionally unsatisfiable:**

$$\{[p_1],\ [\neg p_3],\ [\neg p_1, p_2],\ [\neg p_2, p_3]\}.$$

Example (Cont.)

- We obtain the resolution refutations:

1	$[p_1]$	$[p(a)]$	$[p(a)]$	
2	$[\neg p_1, p_2]$	$[\neg p(a), p(f(a))]$	$[\neg p(X), p(f(X))]$	
3	$[\neg p_2, p_3]$	$[\neg p(f(a)), p(f(f(a)))]$	$[\neg p(Y), p(f(Y))]$	
4	$[\neg p_3]$	$[\neg p(f(f(a)))]$	$[\neg p(f(f(a)))]$	
5	$[\neg p_1, p_3]$	$[\neg p(a), p(f(f(a)))]$	$[\neg p(X_1), p(f(f(X_1)))]$	res(2,3)
6	$[p_3]$	$[p(f(f(a)))]$	$[p(f(f(a)))]$	res(1,5)
7	$[\,]$	$[\,]$	$[\,]$	res(4,6)

- Clauses 2 und 3 in the last refutation are variants.
- After deleting one of them we obtain a refutation for:

$$F = \langle [p(a)],\ [\neg p(X),\ p(f(X))],\ [\neg p(f(f(a)))] \rangle.$$

- In general we can not predict how many copies of a clause are needed to obtain a resolution refutations.

Soundness and Completeness of the Resolution Calculus

- **Theorem 4.60** Let F be a sentence. $\models F$ holds iff $\vdash_r F$ holds.
- **Proof**

$\Leftarrow$ **Soundness** Suppose $\vdash_r F$ holds.

- There is a resolution refutation of G, where G is a clause form of $\neg F$.
- Theorem 4.59: G ist unsatisfiable.
- Theorem 4.32: $\neg F$ is unsatisfiable.
- F is valid, i.e., $\models F$ holds.

$\Rightarrow$ **Completeness** Suppose $\models F$ holds.

- F valid and $\neg F$ unsatisfiable.
- Let G be a clause form of $\neg F$.
- Theorem 4.32: G ist unsatisfiable.
- Theorem 4.59: There exists a resulution refutation for G, i.e., $\vdash_r F$. qed

4.8.3 Compactness Theorem

- **Theorem 4.63** Let $\mathcal{F}$ be a set of first order sentences.
  $\mathcal{F}$ is satisfiable **iff** each finite subset of $\mathcal{F}$ is satisfiable.
- We may try to prove the Theorem directly as in propositional logic.
- **Idea** We use the compactness theorem of propositional logic
  in order to find a simpler proof for Theorem 4.63.
- **Problem** $\mathcal{F}$ may be a countably infinite set of sentences.
  In this chapter, however, all our main results dealt only with a single sentence.
- **Task** We must extend some of our results to sets.

4.8.4 Undecidabilty of First Order Logic

- Is first order logic decidable?
- Yes/No-problems.
  - Is a given word (e. g. “uhu”) a palindrome?
- A class of yes/no-questions is decidable, if there is a procedure which terminates for each question from this class in finite time and answers the question correctly with “yes” or “no”.
- Examples
  - Are words over a given alphabet palindromes?
  - Are propositional formulas over a given alphabet valid?

Semi-Decidability

- Observing the predicate logic resolution calculus we find:
  - If F is valid, then it terminates with "yes".
  - If F is not valid, then it either
    - terminates with "no" or
    - runs forever.
- Such problems are undecidable, but semi-decidable.
- Examples
  - Halting problem of Turing machinses
  - Acceptance problem of two-counter machines
  - Post's correspondence problem

Post's Correspondence Problem

- A **Post's correspondence task** consists of an alphabet Σ containing at least two symbols, a natural number n and two lists of words over Σ both having length n.
- A **Post's correspondence problem** is the question of whether there exists a sequence $(i_1, \ldots, i_k)$ of natural number, where $i_j \leq n$ for all $j \geq 1$, such that the concatenation of the i_1-st, i_2-nd, etc. words taken from the first list and the concatenation of the corresponding words taken from the second list are identical.
- **Example** $\Sigma = \{0,\ 1\}$, $n = 3$, $[010,\ 00,\ 100]$, $[10,\ 000,\ 10]$.
  - $(2,\ 1)$ is a solution:

$$00010\ (=\ 00 \cdot 010 = 000 \cdot 10)$$

  - $(3,\ 2)$ is a solution:

$$10000\ (=\ 100 \cdot 00 = 10 \cdot 000)$$

- **Theorem PKP** Post's corresponding problem is undecidable (Post:46).

The Undecidability of First Order Logic

- **Theorem 4.64** Let $\mathcal{F}$ be a set of first order sentences and G a first order sentence. The question of whether $\mathcal{F} \models G$ holds is undecidable.
- **Proof** We show:
  - Post's correspondence problem is decidable iff first order logic is decidable.
  - Let $P = (\Sigma,\ n,\ [\alpha_1, \ldots, \alpha_n],\ [\beta_1, \ldots, \beta_n])$, where $\Sigma = \{0,\ 1\}$.
  - Let $\mathcal{F} = \{a/0,\ f_0/1,\ f_1/1\}$ and $\mathcal{R} = \{p/2\}$.
  - **Notation** $f_{\sigma_1 \ldots \sigma_m}(X) := f_{\sigma_m}(\ldots(f_{\sigma_1}(X))\ldots)$, where $\sigma_i \in \{0, 1\}$.
  - $$\begin{aligned} F_P \quad = \quad & ((\textstyle\bigwedge_{i=1}^{n} p(f_{\alpha_i}(a), f_{\beta_i}(a)) && (1) \\ & \wedge (\forall X, Y)\,(p(X, Y) \rightarrow \textstyle\bigwedge_{i=1}^{n} p(f_{\alpha_i}(X), f_{\beta_i}(Y)))) && (2) \\ & \rightarrow\ (\exists Z)\, p(Z, Z)) && (3) \end{aligned}$$
  - **To show** P has solution **iff** F_P is valid $\rightsquigarrow$ **Exercise**.
  - Theorem 4.64 follows immediately from Theorem PKP. **qed**