Chapter 1

Many-Sorted Logic

1.1 Syntax

1.1.1 Definition
We fix an enumerable set $\text{Sort}$ of sorts.

1.1.2 Definition
We fix an enumerable set $\text{Var}$ of VARIABLES. Each variable has associated to it a sort. We denote with $\text{Var}_\sigma$ the set of variables of sort $\sigma$. We assume that $\text{Var}_\sigma$ is enumerable, for all sorts $\sigma$.

1.1.3 Definition
We fix an enumerable set $\text{Con}$ of CONSTANT SYMBOLS. Each constant symbol has associated to it a sort. We denote with $\text{Con}_\sigma$ the set of constant symbols of sort $\sigma$. We assume that $\text{Con}_\sigma$ is enumerable, for all sorts $\sigma$.

1.1.4 Definition
We fix an enumerable set $\text{Fun}$ of FUNCTION SYMBOLS. Each function symbol has associated to it an arity of the form $\sigma_1 \times \cdots \times \sigma_n \to \sigma$, where $n \geq 1$ and $\sigma_1, \ldots, \sigma_n, \sigma$ are sorts. We denote with $\text{Fun}_{\sigma_1 \times \cdots \times \sigma_n \to \sigma}$ the set of function symbols of arity $\sigma_1 \times \cdots \times \sigma_n \to \sigma$. We assume that $\text{Fun}_{\sigma_1 \times \cdots \times \sigma_n \to \sigma}$ is enumerable, for all sorts $\sigma_1, \ldots, \sigma_n, \sigma$.

1.1.5 Definition
We fix an enumerable set $\text{Pred}$ of PREDICATE SYMBOLS. Each predicate symbol has associated to it an arity of the form $\sigma_1 \times \cdots \times \sigma_n$, where $n \geq 1$ and $\sigma_1, \ldots, \sigma_n$ are sorts. We denote with $\text{Pred}_{\sigma_1 \times \cdots \times \sigma_n}$ the set of predicate symbols of arity $\sigma_1 \times \cdots \times \sigma_n$. We assume that $\text{Pred}_{\sigma_1 \times \cdots \times \sigma_n}$ is enumerable, for all sorts $\sigma_1, \ldots, \sigma_n$. 
1.1.6 Definition
The equality symbol is $\approx$.

1.1.7 Definition
The propositional connectives are
1. $\neg$ (not);
2. $\land$ (and);
3. $\lor$ (or);
4. $\rightarrow$ (implies);
5. $\leftrightarrow$ (iff).

1.1.8 Definition
The universal quantifier is $\forall$.

1.1.9 Definition
The existential quantifier is $\exists$.

1.1.10 Definition
A signature is a tuple $\Sigma = (S, C, F, P)$ where:
1. $S$ is a nonempty set of sorts.
2. $C$ is a countable set of constant symbols whose sorts belong to $S$.
3. $F$ is a countable set of function symbols whose arities are constructed using sorts that belong to $S$.
4. $P$ is a countable set of predicate symbols whose arities are constructed using sorts that belong to $S$.

Given a signature $\Sigma = (S, C, F, P)$, we write $\Sigma^S$ for $S$, $\Sigma^C$ for $C$, $\Sigma^F$ for $F$, and $\Sigma^P$ for $P$.

1.1.11 Definition
Let $\Sigma$ be a signature. The set of $\Sigma$-terms of sort $\sigma$ is the smallest set of expressions satisfying the following properties:

- Each variable $x$ of sort $\sigma$ is a term of sort $\sigma$, provided that $\sigma \in \Sigma^S$.
- Each constant symbol $c \in \Sigma^C$ of sort $\sigma$ is a $\Sigma$-term of sort $\sigma$.
- If $f \in \Sigma^F$ is a function symbol of arity $\sigma_1 \times \cdots \times \sigma_n \rightarrow \sigma$ and $t_i$ is a $\Sigma$-term of sort $\sigma_i$, for $i = 1, \ldots, n$, then $f(t_1, \ldots, t_n)$ is a term of sort $\sigma$.

1.1.12 Definition
Let $\Sigma$ be a signature. A $\Sigma$-atom is an expression of the form

$$s \approx t, \quad p(t_1, \ldots, t_n),$$

where:
1.1. Syntax

1. $s$ and $t$ are $\Sigma$-terms of the same sort;

2. $p \in \Sigma^P$ is a predicate symbol of arity $\sigma_1 \times \cdots \times \sigma_n$ and $t_i$ is a $\Sigma$-term of sort $\sigma_i$, for $i = 1, \ldots, n$.

1.1.13 Definition

The set of $\Sigma$-formulae is the smallest set of expressions satisfying the following properties:

1. Each $\Sigma$-atom is a $\Sigma$-formula.

2. If $\varphi$ is a $\Sigma$-formula then $\neg \varphi$ is a $\Sigma$-formula.

3. If $\varphi$ and $\psi$ are $\Sigma$-formulae then $\varphi \wedge \psi$, $\varphi \vee \psi$, $\varphi \rightarrow \psi$, and $\varphi \leftrightarrow \psi$ are formulae.

4. If $\varphi$ is a $\Sigma$-formula, $\sigma \in \Sigma^S$, and $x$ is a variable of sort $\sigma$, then $(\forall \sigma x) \varphi$ and $(\exists \sigma x) \varphi$ are $\Sigma$-formulae.

1.1.14 Definition

A $\Sigma$-literal is a formula of the form

$$\varphi, \quad \neg \varphi,$$

where $\varphi$ is a $\Sigma$-atom.

1.1.15 Definition

A quantifier-free $\Sigma$-formula is a $\Sigma$-formula in which no quantifier occurs.

1.1.16 Definition

Let $t$ be a term, and let $\sigma$ be a sort. We denote with $\text{vars}_\sigma(t)$ the set of variables of sort $\sigma$ occurring in $t$. This set can be recursively defined as follows:

1. $\text{vars}_\sigma(x) = \{x\}$, for all variables $x$ of sort $\sigma$.

2. $\text{vars}_\sigma(x) = \emptyset$, for all variables $x$ whose sort is not $\sigma$.

3. $\text{vars}_\sigma(c) = \emptyset$, for all constant symbols $c$.

4. $\text{vars}_\sigma(f(t_1, \ldots, t_n)) = \bigcup_{i=1}^n \text{vars}_\sigma(t_i)$.

1.1.17 Definition

Let $t$ be a term. We denote with $\text{vars}(t)$ the set of variables occurring in $t$, that is,

$$\text{vars}(t) = \bigcup_{\sigma \in \text{Sort}} \text{vars}_\sigma(t).$$

1.1.18 Definition

Let $T$ be a set of terms. We let

$$\text{vars}_\sigma(T) = \bigcup_{t \in T} \text{vars}_\sigma(t),$$
1.1.19 Definition
Let $T$ be a set of terms. We let

$$\text{vars}(T) = \bigcup_{t \in T} \text{vars}(t).$$

1.1.20 Definition
Let $\varphi$ be a formula, and let $\sigma$ be a sort. We denote with $\text{vars}_\sigma(\varphi)$ the set of variables occurring free in $\varphi$. This set can be recursively defined as follows:

1. $\text{vars}_\sigma(s \approx t) = \text{vars}_\sigma(s) \cup \text{vars}_\sigma(t)$.
2. $\text{vars}_\sigma(p(t_1, \ldots, t_n)) = \bigcup_{i=1}^n \text{vars}_\sigma(t_i)$.
3. $\text{vars}_\sigma(\neg \varphi_1) = \text{vars}_\sigma(\varphi_1)$.
4. $\text{vars}_\sigma(\varphi_1 \land \varphi_2) = \text{vars}_\sigma(\varphi_1) \cup \text{vars}_\sigma(\varphi_2)$.
5. $\text{vars}_\sigma(\varphi_1 \lor \varphi_2) = \text{vars}_\sigma(\varphi_1) \cup \text{vars}_\sigma(\varphi_2)$.
6. $\text{vars}_\sigma(\varphi_1 \rightarrow \varphi_2) = \text{vars}_\sigma(\varphi_1) \cup \text{vars}_\sigma(\varphi_2)$.
7. $\text{vars}_\sigma((\forall \tau x)\varphi_1) = \text{vars}_\sigma(\varphi_1) \setminus \{x\}$.
8. $\text{vars}_\sigma((\exists \tau x)\varphi_1) = \text{vars}_\sigma(\varphi_1) \setminus \{x\}$.

1.1.21 Definition
Let $\varphi$ be a formula. We denote with $\text{vars}(\varphi)$ the set of variables occurring free in $\varphi$, that is,

$$\text{vars}(\varphi) = \bigcup_{\sigma \in \text{Sort}} \text{vars}_\sigma(\varphi).$$

1.1.22 Definition
Let $\Phi$ be a set of formulae. We let

$$\text{vars}_\sigma(\Phi) = \bigcup_{\varphi \in \Phi} \text{vars}_\sigma(\varphi),$$

1.1.23 Definition
Let $\Phi$ be a set of formulae. We let

$$\text{vars}(\Phi) = \bigcup_{\varphi \in \Phi} \text{vars}(\varphi).$$

1.1.24 Definition
Let $\Sigma$ be a signature. A $\Sigma$-sentence is a $\Sigma$-formula $\varphi$ such that $\text{vars}(\varphi) = \emptyset$. 
1.2 Semantics

1.2.1 Definition
Let $\Sigma$ be a signature, and let $X$ be a set of variables whose sorts are in $\Sigma^S$. A $\Sigma$-interpretation over $X$ is a map satisfying the following properties:

1. Each sort $\sigma \in \Sigma^S$ is mapped to a nonempty domain $A_{\sigma}$.
2. Each variable $x \in X$ of sort $\sigma$ is mapped to an element $x^A \in A_{\sigma}$.
3. Each constant symbol $c \in \Sigma^C$ of sort $\sigma$ is mapped to an element $c^A \in A_{\sigma}$.
4. Each function symbol $f \in \Sigma^F$ of arity $\sigma_1 \times \cdots \times \sigma_n \rightarrow \sigma$ is mapped to a function $f^A : A_{\sigma_1} \times \cdots \times A_{\sigma_n} \rightarrow A_{\sigma}$.
5. Each predicate symbol $p \in \Sigma^P$ of arity $\sigma_1 \times \cdots \times \sigma_n$ is mapped to a subset $p^A \subseteq A_{\sigma_1} \times \cdots \times A_{\sigma_n}$.

1.2.2 Definition
Let $\Sigma$ be a signature. A $\Sigma$-structure is a $\Sigma$-interpretation over an empty set of variables.

1.2.3 Definition
Let $\Sigma$ be a signature, let $t$ be a $\Sigma$-term of sort $\sigma$, and let $A$ be a $\Sigma$-interpretation over $X$ such that $\text{vars}(t) \subseteq X$. The evaluation of $t$ under $A$ is the object $t^A \in A_{\sigma}$ recursively defined as follows:

1. The evaluation of a variable $x$ is $x^A$.
2. The evaluation of a constant symbol $c$ is $c^A$.
3. The evaluation of a term $f(t_1, \ldots, t_n)$ is
$$\left[f(t_1, \ldots, t_n)\right]^A = f^A(t_1^A, \ldots, t_n^A).$$

1.2.4 Definition
Let $A$ and $B$ be $\Sigma$-interpretations over $X$, and let $x \in X$ be a variable. We say that $B$ is an $x$-variant of $A$ if:

1. $A_{\sigma} = B_{\sigma}$, for all sorts $\sigma \in \Sigma^S$.
2. $r^A = r^B$, for all objects $r \in \Sigma^C \cup \Sigma^F \cup \Sigma^P \cup (X \setminus \{x\})$.

1.2.5 Definition
Let $\Sigma$ be a signature, let $\varphi$ be a $\Sigma$-formula, and let $A$ be a $\Sigma$-interpretation over $X$ such that $\text{vars}(\varphi) \subseteq X$. The evaluation of $\varphi$ under $A$ is the truth value $\varphi^A \in A_{\sigma}$ recursively defined as follows:

1. $[s \approx t]^A = \text{true} \iff s^A = t^A$.
2. $[p(t_1, \ldots, t_n)]^A = \text{true} \iff (t_1^A, \ldots, t_n^A) \in p^A$. 


3. $[-\varphi]^A = true \iff \varphi^A = false$.
4. $[\varphi \land \psi]^A = true \iff \varphi^A = true$ and $\psi^A = true$.
5. $[\varphi \lor \psi]^A = true \iff \varphi^A = true$ or $\psi^A = true$.
6. $[\varphi \rightarrow \psi]^A = true \iff \varphi^A = false$ or $\psi^A = true$.
7. $[(\forall x)\varphi]^A = true \iff \varphi^B = true$, for all $x$-variants $B$ of $A$.
8. $[(\exists x)\varphi]^A = true \iff \varphi^B = true$, for some $x$-variant $B$ of $A$.

1.2.6 Definition
Let $A$ be a $\Sigma$-interpretation over $X$, and let $\varphi$ be a $\Sigma$-formula such that $\text{vars}(\varphi) \subseteq X$. We write $A \models \varphi$
when $\varphi^A = true$.

1.2.7 Definition
Let $\varphi$ be a $\Sigma$-formula, and let $X = \text{vars}(\varphi)$. We say that $\varphi$ is:
- **valid**, if $A \models \varphi$, for all $\Sigma$-interpretations $A$ over $X$;
- **satisfiable**, if $A \models \varphi$, for some $\Sigma$-interpretation $A$ over $X$;
- **unsatisfiable**, if $\varphi$ is not satisfiable.

1.2.8 Definition
Let $A$ be a $\Sigma$-interpretation over $X$, and let $\Phi$ be a set of $\Sigma$-formulae such that $\text{vars}(\Phi) \subseteq X$. We write $A \models \Phi$
when $A \models \varphi$, for all formulae $\varphi \in \Phi$.

1.2.9 Definition
Let $\Phi$ be a set of $\Sigma$-formulae, and let $X = \text{vars}(\Phi)$. We say that $\Phi$ is:
- **valid**, if $A \models \Phi$, for all $\Sigma$-interpretations $A$ over $X$;
- **satisfiable**, if $A \models \Phi$, for some $\Sigma$-interpretation $A$ over $X$;
- **unsatisfiable**, if $\Phi$ is not satisfiable.
1.3. Modelclasses

1.2.10 Definition
Let $A$ be a $\Sigma$-interpretation over $X$. For $\Sigma_0 \subseteq \Sigma$ and $X_0 \subseteq X$, we denote with $A^{\Sigma_0, X_0}$ the interpretation obtained from $A$ by restricting it to interpret only the symbols in $\Sigma_0$ and the variables in $X_0$. Furthermore, we let $A^{\Sigma_0, \emptyset} = A^{\Sigma_0, \emptyset}$.

1.2.11 Definition
Let $A$ and $B$ be two $\Sigma$-interpretations over $X$. An isomorphism $h$ of $A$ into $B$ is a family of bijective functions

$$h = \{h_\sigma : A_\sigma \to B_\sigma \mid \sigma \in \Sigma^S\}$$

such that:

1. $h_\sigma(x^A) = x^B$, for all variables $x \in X_\sigma$.
2. $h_\sigma(c^A) = c^B$, for all constant symbols $c \in \Sigma^C$.
3. $h_\sigma(f^A(a_1, \ldots, a_n)) = f^B(h_\sigma_1(a_1), \ldots, h_\sigma_n(a_n))$, for all function symbol $f \in \Sigma^F$ of arity $\sigma_1 \times \cdots \times \sigma_n \rightarrow \sigma$.
4. $(a_1, \ldots, a_n) \in p^A$ if and only if $(h_\sigma_1(a_1), \ldots, h_\sigma_n(a_n)) \in p^B$, for all predicate symbol $p \in \Sigma^P$ of arity $\sigma_1 \times \cdots \times \sigma_n$.

We write $A \cong B$ when there is an isomorphism of $A$ into $B$.

1.3 Modelclasses

1.3.1 Definition
A $\Sigma$-MODELCLASS is a pair $M = (\Sigma, A)$ such that:

1. $\Sigma$ is a signature;
2. $A$ is a class of $\Sigma$-structures;
3. $A$ is closed under isomorphism.

1.3.2 Definition
Let $M = (\Sigma, A)$ be a modelclass. An $M$-STRUCTURE is a $\Sigma$-structure $A$ such that $A \in A$.

1.3.3 Definition
Let $M = (\Sigma, A)$ be a modelclass. An $M$-INTERPRETATION is a $\Sigma$-interpretation $A$ such that $A^{\Sigma}$ is a $\Sigma$-structure.

1.3.4 Definition
Let $M$ be a $\Sigma$-modelclass, let $A$ be a $\Sigma$-interpretation over $X$, and let $\varphi$ be a $\Sigma$-formula such that $\text{vars}(\varphi) \subseteq X$. We write

$$A \models_M \varphi,$$

whenever $\varphi^A = \text{true}$ and $A^{\Sigma}$ is an $M$-structure.
1.3.5 Definition
Let $M$ be a $\Sigma$-modelclass, let $\varphi$ be a $\Sigma$-formula, and let $X = \text{vars}(\varphi)$. We say that $\varphi$ is:
- $M$-valid, if $A \models_M \varphi$, for all $M$-interpretations $A$ over $X$;
- $M$-satisfiable, if $A \models_M \varphi$, for some $M$-interpretation $A$ over $X$;
- $M$-unsatisfiable, if $\varphi$ is not $M$-satisfiable.

1.3.6 Definition
Let $M$ be a $\Sigma$-modelclass, let $A$ be a $\Sigma$-interpretation over $X$, and let $\Phi$ be a set of $\Sigma$-formulae such that $\text{vars}(\Phi) \subseteq X$. We write
$$A \models_M \Phi$$
when
$$A \models_M \varphi, \quad \text{for all formulae } \varphi \in \Phi.$$

1.3.7 Definition
Let $M$ be a $\Sigma$-modelclass, let $\Phi$ be a set of $\Sigma$-formulae, and let $X = \text{vars}(\Phi)$. We say that $\Phi$ is:
- $M$-valid, if $A \models_M \Phi$, for all $\Sigma$-interpretations $A$ over $X$;
- $M$-satisfiable, if $A \models_M \Phi$, for some $\Sigma$-interpretation $A$ over $X$;
- $M$-unsatisfiable, if $\Phi$ is not $M$-satisfiable.

1.3.8 Definition
Let $M$ be a $\Sigma$-modelclass, and let $L$ be a set of $\Sigma$-formulae. We define the following decision problems:
- The validity problem of $M$ with respect to $L$ is the problem of deciding, for each $\Sigma$-formula $\varphi \in L$, whether or not $\varphi$ is $M$-valid.
- The satisfiability problem of $M$ with respect to $L$ is the problem of deciding, for each $\Sigma$-formula $\varphi \in L$, whether or not $\varphi$ is $M$-satisfiable.
- The unsatisfiability problem of $M$ with respect to $L$ is the problem of deciding, for each $\Sigma$-formula $\varphi \in L$, whether or not $\varphi$ is $M$-unsatisfiable.

When we mention a decision problem without specifying the set of formulae $L$, we implicitly assume that $L$ is the set of all $\Sigma$-formulae. For instance, if $M$ is a $\Sigma$-modelclass, the validity problem of a $\Sigma$-model class $M$ is the problem of deciding, for each $\Sigma$-formula $\varphi$ whether or not $\varphi$ is $M$-valid.

When we prefix the name of a decision problem with “quantifier-free”, we implicitly assume that $L$ is the set of all quantifier-free $\Sigma$-formulae. For instance, the quantifier-free satisfiability problem of a $\Sigma$-model class $M$ is the problem of deciding, for each quantifier-free $\Sigma$-formula $\varphi$ whether or not $\varphi$ is $M$-satisfiable.