We now determine which objects are described by terms and when a formula is true or false.

**Interpretation:**
- determines a set of objects that we speak about (carrier \( A \))
- maps every function symbol \( f \) to a function \( \alpha_f \)
- maps every predicate symbol \( p \) to a relation \( \alpha_p \)
- assigns objects to variables (variable assignment \( \beta \))

**Def 2.2.1 (Interpretation, Structure)**

For a signature \((\Sigma, \Delta)\), an interpretation has the form \( I = (A, \alpha, \beta) \).

- \( A \) is an arbitrary set with \( A \neq \emptyset \) (carrier).
- \( \alpha \) maps every \( f \in \Sigma_n \) to a function \( \alpha_f : A \times \cdots \times A \rightarrow A \) \( n \) times.
- \( \alpha \) maps every \( p \in \Delta_n \) to a \( \alpha_p \subseteq \underbrace{A \times \cdots \times A}_{n \ times} \) if \( n \geq 1 \).

For \( p \in \Delta_0 \), \( \alpha_p \in \{ \text{TRUE}, \text{FALSE} \} \).

We say that \( \alpha_f \) resp. \( \alpha_p \) is the meaning of \( f \) resp. \( p \).

\( \beta : \mathcal{V} \rightarrow A \) is called a variable assignment.

Every interpretation gives a meaning to every term, i.e. \( I : \mathcal{V}(\Sigma, \mathcal{V}) \rightarrow A \) as follows:
\[ I(X) = \beta(X) \]
\[ I(f(t_1, \ldots, t_n)) = \alpha_f(I(t_1), \ldots, I(t_n)) \]

**Ex. 2.2.2 Example Interpretation** \( I = (A, \alpha, \beta) \) with
\[ A = \mathbb{N} \]
\[ \alpha_n = n \text{ for all } n \in \mathbb{N} \]
\[ \alpha_{\text{monica}} = 0 \]
\[ \alpha_{\text{karin}} = 1 \]
\[ \alpha_{\text{vena}} = 2 \]
\[ \vdots \]
\[ \alpha_{\text{date}}(n_1, n_2, n_3) = n_1 + n_2 + n_3 \]
\[ \beta(\text{male}) = \{ n \mid n \text{ is even} \} \]
\[ \beta(\text{female}) = \{ n \mid n \text{ is odd} \} \]
\[ \beta(\text{human}) = \mathbb{N} \]
\[ \beta(\text{married}) = \{ (n, m) \mid n > m \} \]
\[ \beta(X) = 0 \]
\[ \beta(Y) = 1 \]
\[ \beta(Z) = 2 \]

Then \( I(\text{date}(X, X, \text{karin})) = \alpha_{\text{date}}(\alpha_X, \beta(X), \alpha_{\text{karin}}) \)
\[ = 1 + 0 + 1 = 2 \]

**Def 2.2.1 (cont.)**
For \( X \in \mathcal{V} \) and \( a \in A \), let \( \beta[X/a] \) be the variable assignment with \( \beta[X/a](X) = a \) and \( \beta[X/a](Y) = \beta(Y) \) for all \( Y \in \mathcal{V} \) with \( Y \neq X \).
For \( I = (A, \alpha, \beta) \), let \( I[X/a] = (A, \alpha, \beta[X/a]) \).

An interpretation \( I = (A, \alpha, \beta) \) satisfies a formula \( \varphi \in \mathcal{F}(\Sigma, \mathcal{A}, \mathcal{V}) \), denoted \( " I \models \varphi " \), if
- \( \varphi = \varphi(t_1, \ldots, t_n) \) and \( (I(t_1), \ldots, I(t_n)) \in \alpha_\varphi \)
\[ \psi(p(t_1, \ldots, t_n)) \text{ and } (I(t_1), \ldots, I(t_n)) \in \alpha \]

\[ \text{if } p \in \Delta_n \text{ and } n \geq 1 \]

\[ \psi = \varphi \text{ and } \alpha \varphi = \text{TRUE} \text{ if } p \in \Delta_0 \]

\[ \psi = \neg \psi_1 \text{ and } I \not\models \psi_1 \]

\[ \psi = \psi_1 \lor \psi_2 \text{ and } I \models \psi_1 \text{ and } I \not\models \psi_2 \]

\[ \psi = \psi_1 \lor \psi_2 \text{ and } (I \models \psi_1 \text{ or } I \models \psi_2) \]

\[ \psi = \psi_1 \Rightarrow \psi_2 \text{ and if } I \models \psi_1 \text{ then also } I \models \psi_2 \]

\[ \psi = \psi_1 \Leftarrow \psi_2 \text{ and } (I \models \psi_1 \text{ iff } I \models \psi_2) \]

\[ \psi = \forall X \psi_1 \text{ and } I[\forall X/a] \models \psi_1 \text{ for all } a \in A \]

\[ \psi = \exists X \psi_1 \text{ and there exists } a \in A \text{ and that } I[\exists X/a] \models \psi_1 \]

\[ \text{Ex 2.2.2 (cont.)} \]

\[ I \models \text{married}(\text{date}(1, X, \text{Karim}), \text{Karim}), \text{ because } \]

\[ (I(\text{date}(1, X, \text{Karim})), I(\text{Karim})) \in \alpha \text{married} \]

\[ \alpha \text{married} = 1 \{ (n, m) \mid n > m \} \]

\[ I \models \forall X \text{ female}(\text{date}(X, X, \text{monika})), \text{ because } \]

\[ I[\forall X/a] \models (\text{date}(X, X, \text{monika})) \in \alpha \text{female} \]

\[ \alpha \text{female} = 1 \{ n \mid n \text{ is even} \} \]
\( \alpha + \alpha + 0 \)

holds for all \( \alpha \in \mathbb{N} \).

**Def 22.1 (cont.)**

An interpretation \( I \) is a model of \( \varphi \) iff \( I \models \varphi \).

\( I \) is a model of a set of formulas \( \Phi \) iff

\( I \models \varphi \) for all \( \varphi \in \Phi \).

Two formulas \( \varphi_1, \varphi_2 \) are equivalent iff we have

\( (I \models \varphi_1 \iff I \models \varphi_2) \) for all interpretations \( I \).

A formula (resp. a set of formulas) is **satisfiable** iff it has a model.

A formula is **valid** if every interpretation is a model of the formula. (tautology (general validity))

An interpretation without variable assignment is called a **structure** \( S = (A, \alpha) \).

If we only regard closed formulas, then satisfiability etc. can be defined for structures:

\( S \models \varphi \) iff \( I \models \varphi \) for some interpretation

\[ \text{married}(gerd, eros) \]

is satisfiable, not valid

\[ \text{married}(2, 1) \lor \neg \text{married}(2, 1) \]

is valid

\[ \varphi \lor \neg \varphi \]

is unsatisfiable for any formula \( \varphi \).
\[ I = (\mathcal{A}, \alpha, \beta) \]

Similarly, we can define \( S(t) \) for ground terms \( t \).

Two similar concepts:

Substitution \( \sigma : \mathcal{V} \rightarrow \mathcal{T}(\Sigma, \mathcal{V}) \) (syntactic)

Variable assignment \( \beta : \mathcal{V} \rightarrow \mathcal{A} \) (semantic)

**Lemma 2.2.3 (Substitution Lemma)**

Let \( I = (\mathcal{A}, \alpha, \beta) \) be an interpretation for a signature \((\Sigma, \Delta)\),

let \( \sigma = \{X_1/t_1, \ldots, X_n/t_n\} \). Then we have

(a) \( I(\sigma(t)) = I[\prod_{i} X_i/I(t_i), \ldots, X_n/I(t_n)](t) \) for all \( t \in \mathcal{T}(\Sigma, \mathcal{V}) \)

(b) \( I \vdash \sigma(\psi) \) if and only if \( \prod_{i} X_i/I(t_i), \ldots, X_n/I(t_n) \models \psi \)

for all \( \psi \in \mathcal{T}(\Sigma, \mathcal{V}) \).

**Exercise 2.24** Let \( I \) be the interpreter from Exercise 2.2.2.

Let \( \sigma = \{X/\text{date}(1, X, \text{Karin}), Y/t\} \), let \( t = \text{date}(X, Y, Z) \).

\[
I(\sigma(t)) = I(\text{date}(\text{date}(1, X, \text{Karin}), Y, t)) \\
= \lambda \text{date} (\lambda \text{date} (\lambda \text{date}(\alpha, \beta(X), \alpha_{\text{Karin}}), \beta(Y), \beta(Z))) \\
= 1 + 0 + 1 + 1 + 2 = 5
\]

\[
I[\prod_{i} X_i/(\text{date}(1, X, \text{Karin})) = (\text{date}(X, Y, Z)](t) = \frac{2}{2} \lambda \text{date} (\beta[\prod_{i} X_i/2\pi(X)], \beta[\prod_{i} X_i/2\pi(Y)], \beta[\prod_{i} X_i/2\pi(Z)]) = 2 + 1 + 2 = 5
\]
Proof of Lemma 2.2.3 (Substitution Lemma)

\[ I(\sigma(t)) = I[I[t_1/\sigma(t_1)], \ldots, X_n/\sigma(t_n)](t) \]
for all terms \( t \).

Proof by structural induction on \( t \) (Induction corresponds to the definition of the data structure for terms).

Ind. Base: prove the statement for the smallest possible terms (i.e. \( t \in U \cup \Sigma_0 \))

Ind. Step: prove the statement for terms of the form \( f(S_{n_1}, \ldots, S_n) \)

Ind. Hypothesis: statement already holds for smaller terms \( S_{n_1}, \ldots, S_n \).

Ind. Base \( t \) is a variable (The case where \( t \in \Sigma_0 \) is handled together with the ind. step)

\( \sigma = \{X_1/t_1, \ldots, X_n/t_n\} \).

Case 1: \( t = X_i \)
\[ I(\sigma(X_i)) = I(t_i) \]
\[ I[I[X_i/I(t_1), \ldots, X_n/I(t_n)](X_i)] = I(t_i) \]
**Case 2:** \( t \) is a variable \( Y \notin \{x_1, ..., x_n\} \\
I(\sigma(Y)) = I(Y) = \beta(Y) \\
I[I[x_n/I(t_n), ..., x_n/I(t_n)](Y) = \beta[I[x_n/I(t_n), ..., x_n/I(t_n)](Y) \\
= \beta(Y)\\

**Ind. Step:** \( t = f(s_n, ..., s_k) \) with \( k \geq 0 \\
I(\sigma(t)) = I(\sigma(f(s_n, ..., s_k))) = I(f(\sigma(s_n), ..., \sigma(s_k))) \\
= \chi_f(I(\sigma(s_n)), ..., I(\sigma(s_k)))\\

\[
I[I[x_n/I(t_n), ..., x_n/I(t_n)](t) = \\
\chi_f[I[I[x_n/I(t_n), ..., x_n/I(t_n)](s_n), ..., I[I[x_n/I(t_n), ..., x_n/I(t_n)](s_k)]
\]

The proof of (b) is similar, by structural induction on formulas.

**Def 225 (Entailment)**

A set of formulas \( \Phi \) entails a formula \( \varphi \) (denoted \( \Phi \models \varphi \))
\( \Phi \models \psi \) iff for all interpretations \( I \) with
\( I \models \Phi \) we also have \( I \models \psi \).

(If \( \Phi \) and \( \psi \) have no free variables, then
\( \Phi \models \psi \) means that \( S \models \Phi \) implies \( S \models \psi \)
for all structures \( S \).

So "\( \models \)" has two meanings:
\( I \models \psi \) means: the interpr. \( I \) satisfies \( \psi \)
\( \Phi \models \psi \) means: \( \Phi \) entails \( \psi \)

Instead of \( \emptyset \models \psi \), we also write \( \models \psi \).

Ex. 2.2.6
Let \( \Phi \) be the set of formulas from Ex. 2.1.7 that
corresponds to the logic prog. from Chapter 1.

Asking the query
\(? \leftarrow \text{male (gerd)} \).

means that one has to prove
\( \Phi \models \text{male (gerd)} \).

This clearly holds, because \( \text{male (gerd)} \in \Phi \).

Asking \(? \leftarrow \text{human (gerd)} \).
Asking $\neg \text{human (gerd)}$. means that one has to prove

$\emptyset \models \text{human (gerd)}$.

This holds, as $\emptyset$ contains $\forall X \text{human (X)}$:

$I \models \forall X \text{human (X)}$ $

I \models \forall X /a I \models \text{human (X)}$ for all $a \in A$ $

I \models X /a \models \text{human (X)}$ by the Subst. Lemma

$I \models \text{human (gerd)}$

Asking the query $\neg \text{motherOf (X, susanne)}$. means that one has to prove

$\emptyset \models \exists X \text{motherOf (X, susanne)}$

How does Prolog do this automatically?