9. Interpretations and Assignments

We refer to a subset $\mathcal{L} \subseteq \mathcal{L}^{FOPC}$ containing all the logical symbols, but possibly only some non-logical as a **language** (or **first-order language**).

9.1 Definition Let $\mathcal{L}$ be a language. An **interpretation** of $\mathcal{L}$ is an $\mathcal{L}$-**structure** $\mathcal{A} :=$

$< A; (f_{\mathcal{A}})_{f \in \mathsf{Fct}(\mathcal{L})}; (P_{\mathcal{A}})_{P \in \mathsf{Pred}(\mathcal{L})}; (c_{\mathcal{A}})_{c \in \mathsf{Const}(\mathcal{L})} >$,

i.e.
- A is a non-empty set, the **domain** of $\mathcal{A}$,
- for each k-ary function symbol $f = f_n^{(k)} \in \mathcal{L}$, $f_{\mathcal{A}} : A^k \to A$ is a function
- for each k-ary predicate symbol $P = P_n^{(k)} \in \mathcal{L}$, $P_{\mathcal{A}}$ is a k-ary relation on A, i.e. $P_{\mathcal{A}} \subseteq A^k$ (write $P_{\mathcal{A}}(a_1, \ldots, a_k)$ for $(a_1, \ldots, a_k) \in P_{\mathcal{A}}$)
- for each $c \in$ Const($\mathcal{L}$): $c_{\mathcal{A}} \in A$.

9.2 Definition

Let $\mathcal{L}$ be a language and let $\mathcal{A} =< A; \ldots >$ be an $\mathcal{L}$-structure.

(1) An **assignment** in $\mathcal{A}$ is a function

$$v : \{x_0, x_1, \ldots\} \to A$$

(2) v determines an assignment

$$\widetilde{v} = \widetilde{v}_{\mathcal{A}} : \text{Terms}(\mathcal{L}) \to A$$

defined recursively as follows:
(i) $\widetilde{v}(x_i) = v(x_i)$ for all $i = 0, 1, \ldots$
(ii) $\widetilde{v}(c) = c_{\mathcal{A}}$ for each $c \in \text{Const}(\mathcal{L})$
(iii) $\widetilde{v}(f(t_1, \ldots, t_k)) = f_{\mathcal{A}}(\widetilde{v}(t_1), \ldots, \widetilde{v}(t_k))$ for each $f = f_n^{(k)} \in \text{Fct}(\mathcal{L})$, where the $\widetilde{v}(t_i)$ are already defined.

(3) v determines a **valuation**

$$\widetilde{v} = \widetilde{v}_{\mathcal{A}} : \text{Form}(\mathcal{L}) \to \{T, F\}$$

as follows:

(i) for atomic formulas $\phi \in \mathrm{Form}(\mathcal{L})$:
- for each $P = P_n^{(k)} \in \mathrm{Pred}(\mathcal{L})$ and for all $t \in \mathrm{Term}(\mathcal{L})$

$$\widetilde{v}(P(t_1, \ldots, t_k)) = \begin{cases} T & \text{if } P_{\mathcal{A}}(\widetilde{v}(t_1), \ldots, \widetilde{v}(t_k)) \\ F & \text{otherwise} \end{cases}$$

- for all $t_1, t_2 \in \mathrm{Term}(\mathcal{L})$:

$$\widetilde{v}(t_1 \doteq t_2) = \begin{cases} T & \text{if } \widetilde{v}(t_1) = \widetilde{v}(t_2) \\ F & \text{otherwise} \end{cases}$$

(ii) for arbitrary formulas $\phi \in \mathrm{Form}(\mathcal{L})$ recursively:
- $\widetilde{v}(\neg\psi) = T$ iff $\widetilde{v}(\psi) = F$
- $\widetilde{v}(\psi \to \chi) = T$ iff $\widetilde{v}(\psi) = F$ or $\widetilde{v}(\chi) = T$
- $\widetilde{v}(\forall x_i \psi) = T$ iff $\widetilde{v}^\star(\psi) = T$ for all assignments $v^\star$ agreeing with v except possibly at x_i.

Notation: Write $\mathcal{A} \models \phi[v]$ for $\widetilde{v}_{\mathcal{A}}(\phi) = T$, and say '$\phi$ *is true in* $\mathcal{A}$ *under the assignment* $v = v_{\mathcal{A}}$.'

9.3 Some abbreviations

We use ...	as abbreviation for ...
$(\alpha \vee \beta)$	$((\alpha \to \beta) \to \beta)$
$(\alpha \wedge \beta)$	$\neg(\neg\alpha \vee \neg\beta)$
$(\alpha \leftrightarrow \beta)$	$((\alpha \to \beta) \wedge (\beta \to \alpha))$
$\exists x_i \phi$	$\neg\forall x_i \neg\phi$

9.4 Lemma

For any $\mathcal{L}$-structure $\mathcal{A}$ and any assignment v in $\mathcal{A}$ one has

$$\begin{array}{rll} \mathcal{A} \models (\alpha \vee \beta)[v] & \text{iff} & \mathcal{A} \models \alpha[v] \text{ or } \mathcal{A} \models \beta[v] \\ \mathcal{A} \models (\alpha \wedge \beta)[v] & \text{iff} & \mathcal{A} \models \alpha[v] \text{ and } \mathcal{A} \models \beta[v] \\ \mathcal{A} \models (\alpha \leftrightarrow \beta)[v] & \text{iff} & \widetilde{v}(\alpha) = \widetilde{v}(\beta) \\ \mathcal{A} \models \exists x_i \phi[v] & \text{iff} & \text{for some assignment} \\ & & v^\star \text{ agreeing with } v \\ & & \text{except possibly at } x_i \\ & & \mathcal{A} \models \phi[v^\star] \end{array}$$

Proof: easy

9.5 Example

Let f be a binary function symbol, let '$\mathcal{L} = \{f\}$' (need only list non-logical symbols), consider $\mathcal{A} =< \mathbf{Z}; \cdot >$ as $\mathcal{L}$-structure, let v be the assignment $v(x_i) = i(\in \mathbf{Z})$ for $i = 0, 1, \ldots$, and let

$$\phi = \forall x_0 \forall x_1 (f(x_0, x_2) \doteq f(x_1, x_2) \to x_0 \doteq x_1)$$

Then

$$
\begin{array}{ll}
 & \mathcal{A} \models \phi[v] \\
\text{iff} & \text{for all } v^\star \text{ with } v^\star(x_i) = i \text{ for } i \neq 0 \\
 & \mathcal{A} \models \forall x_1 (f(x_0, x_2) \doteq f(x_1, x_2) \to x_0 \doteq x_1)[v^\star] \\
\text{iff} & \text{for all } v^{\star\star} \text{ with } v^{\star\star}(x_i) = i \text{ for } i \neq 0, 1 \\
 & \mathcal{A} \models (f(x_0, x_2) \doteq f(x_1, x_2) \to x_0 \doteq x_1)[v^{\star\star}] \\
\text{iff} & \text{for all } v^{\star\star} \text{ with } v^{\star\star}(x_i) = i \text{ for } i \neq 0, 1 \\
 & v^{\star\star}(x_0) \cdot v^{\star\star}(x_2) = v^{\star\star}(x_1) \cdot v^{\star\star}(x_2) \\
 & \text{implies } v^{\star\star}(x_0) = v^{\star\star}(x_1) \\
\text{iff} & \text{for all } a, b \in \mathbf{Z},\ a \cdot 2 = b \cdot 2 \text{ implies } a = b, \\
 & \text{which is true.}
\end{array}
$$

So $\mathcal{A} \models \phi[v]$

However, if $v'(x_i) = 0$ for all i, then would have finished with
... iff for all $a, b \in \mathbf{Z}$, $a \cdot 0 = b \cdot 0$ implies $a = b$, which is false. So $\mathcal{A} \not\models \phi[v']$.

9.6 Example

Let P be a unary predicate symbol, $\mathcal{L} = \{P\}$, $\mathcal{A}$ an $\mathcal{L}$-structure, v any assignment in $\mathcal{A}$, and

$$\phi = ((\forall x_0 P(x_0) \to P(x_1)).$$

Then $\mathcal{A} \models \phi[v]$.

Proof:
$\mathcal{A} \models \phi[v]$ iff
$\mathcal{A} \models \forall x_0 P(x_0)[v]$ implies $\mathcal{A} \models P(x_1)[v]$.

Now suppose $\mathcal{A} \models \forall x_0 P(x_0)[v]$. Then for all $v^\star$ which agree with v except possibly at x_0, $P(x_0)[v^\star]$.

In particular, for $v^\star(x_i) = \begin{cases} v(x_i) & \text{if } i \neq 0 \\ v(x_1) & \text{if } i = 0 \end{cases}$
we have $P_{\mathcal{A}}(v^\star(x_0))$, and hence $P_{\mathcal{A}}(v(x_1))$, i.e. $P(x_1)[v]$.

9.7 Definition

Let $\mathcal{L}$ be any first-order language.

- An $\mathcal{L}$-formula ϕ is **logically valid** ('$\models \phi$') if $\mathcal{A} \models \phi[v]$ for *all* $\mathcal{L}$-structures $\mathcal{A}$ and for *all* assignments v in $\mathcal{A}$.

- $\phi \in \mathrm{Form}(\mathcal{L})$ is **satisfiable** if $\mathcal{A} \models \phi[v]$ for *some* $\mathcal{L}$-structure $\mathcal{A}$ and for *some* assignment v in $\mathcal{A}$.

- For $\Gamma \subseteq \mathrm{Form}(\mathcal{L})$ and $\phi \in \mathrm{Form}(\mathcal{L})$, ϕ is a **logical consequence** of Γ ('$\Gamma \models \phi$') if for *all* $\mathcal{L}$-structures $\mathcal{A}$ and for *all* assignments v in $\mathcal{A}$ with $\mathcal{A} \models \psi[v]$ for all $\psi \in \Gamma$, also $\mathcal{A} \models \phi[v]$.

- $\phi, \psi \in \mathrm{Form}(\mathcal{L})$ are **logically equivalent** if $\{\phi\} \models \psi$ and $\{\psi\} \models \phi$.

Example: $\models \phi$ for ϕ from 9.6

Note:
The symbol '$\models$' is now used in two ways:

$\Gamma \models \phi$ – ϕ is a logical consequence of Γ

$\mathcal{A} \models \phi[v]$ – ϕ is satisfied in the $\mathcal{L}$-structure $\mathcal{A}$ under the assignment v

This shouldn't give rise to confusion, since it will always be clear from the context whether there is a set Γ of $\mathcal{L}$-formulas or an $\mathcal{L}$-structure $\mathcal{A}$ in front of '$\models$'.