3 Structures and Specifications

Terms provide us with a formal specification language for set-theoretic structures. In this language, a specification is a set of equations, and a structure satisfies a specification if it satisfies each of its equations. The idea is well-known from algebra: The axioms for groups are a specification, and the groups are the structures satisfying this specification.

3.1 Evaluation

We start with the evaluation of terms. As an example, consider the term $x + 3$. It evaluates to 5 if x takes the value 2 and the names + and 3 take the values the symbols + and 3 suggest. The example tells us that the evaluation of a term requires a function that assigns values to names. We call such functions *interpretations* and define them as follows.

An **interpretation** is a function $\mathcal{I}$ such that:

1. $\mathit{Dom}\,\mathcal{I} = \mathit{Ty} \cup \mathit{Con} \cup \mathit{Var}$
2. $\mathcal{I}u \in \mathcal{I}(\tau u)$
3. $\mathcal{I}(S \to T) = \{\, f \mid f \text{ function } \mathcal{I}S \to \mathcal{I}T \,\}$

We require interpretations to be defined on all types, on all constants, and all variables since this is convenient and serves the purpose. Condition (3) ensures that functional types are interpreted as one would expect. Thus we know how an interpretation behaves on functional types if we know how it behaves on sorts. Conditation (2) says that the values of constants and variables must be taken from the interpretation of their types.

Proposition 3.1 (Coincidence) If $\mathcal{I}$ and $\mathcal{I}'$ agree on all names, then $\mathcal{I} = \mathcal{I}'$.

Proof We need to show: $\forall T\colon\ \mathcal{I}T = \mathcal{I}'T$. This can be done by induction on $|T|$. ∎

Proposition 3.2 $\mathcal{I}T \neq \emptyset$.

Proof Let $\mathcal{I}$ be an interpretation and T be a type. By Axiom Inf we know that there is a variable x with $\tau x = T$. Hence $\mathcal{I}x \in \mathcal{I}(\tau x) = \mathcal{I}T$. ∎

Given an interpretation $\mathcal{I}$, a variable x and a value $v \in \mathcal{I}(\tau x)$, we use $\mathcal{I}_{x,v}$ to denote the interpretation $\mathcal{I}[x{:=}v]$. Note that $\mathcal{I}_{x,v}$ satisfies the following equations:

$$\mathcal{I}_{x,v}T = \mathcal{I}T$$
$$\mathcal{I}_{x,v}u = \mathit{if}\ u = x\ \mathit{then}\ v\ \mathit{else}\ \mathcal{I}u$$

Proposition 3.3 (Evaluation) For every interpretation $\mathcal{I}$ there exists one and only one function $\hat{\mathcal{I}}$ such that:

1. $Dom\,(\hat{\mathcal{I}}) = Ter$
2. $\hat{\mathcal{I}}t \in \mathcal{I}(\tau t)$
3. $\hat{\mathcal{I}}u = \mathcal{I}u$
4. $\hat{\mathcal{I}}(st) = (\hat{\mathcal{I}}s)(\hat{\mathcal{I}}t)$
5. $\hat{\mathcal{I}}(\lambda x.t) = \lambda v{\in}\mathcal{I}(\tau x).\, \hat{\mathcal{I}}_{x,v}t$

We call $\hat{\mathcal{I}}$ the **evaluation function** for $\mathcal{I}$.

Proof To show the existence of $\hat{\mathcal{I}}$, we define $\hat{\mathcal{I}}$ recursively according to (3), (4) and (5), where (5) is modified such that it requires $x = \varphi(\lambda x.t)$. The properties (1), (2) and the uniqueness of $\hat{\mathcal{I}}$ are immediate consequences of this definition. The unmodified version of (5) can be shown with the Proposition 3.4 whose proof can be based on our definition of $\hat{\mathcal{I}}$. ∎

Given an interpretation $\mathcal{I}$ and a substitution θ, we use $\mathcal{I}_\theta$ to denote the interpretation defined as follows:

$$\mathcal{I}_\theta T = \mathcal{I}T$$

$$\mathcal{I}_\theta u = \hat{\mathcal{I}}(\theta u)$$

Proposition 3.4 (Substitution) $\hat{\mathcal{I}}(\mathsf{S}\theta t) = \hat{\mathcal{I}}_\theta t$.

Proof By induction on $|t|$. Tedious. ∎

Proposition 3.5 (Coincidence) If $\mathcal{I}$ and $\mathcal{I}'$ agree on $\mathcal{N}t$, then $\hat{\mathcal{I}}t = \hat{\mathcal{I}}'t$.

3.2 Signatures and Structures

When we use terms as specification language, we consider only certain sorts and certain constants. A collection of relevant sorts and constants will be called a *signature*, and an interpretation for the names of a signature will be called a *structure*. The precise definitions are as follows.

A **signature** is a set $\Sigma \subseteq Sor \cup Con$ such that $\forall c \in \Sigma\colon\ \mathcal{N}(\tau c) \subseteq \Sigma$. Note that we require that a signature is closed in the sense that if it contains a constant, it must also contain the sorts in the type of the constant.

A **structure** is a function $\mathcal{A}$ such that $Dom\ \mathcal{A}$ is a signature and there exists an interpretation $\mathcal{I}$ such that $\mathcal{A} \subseteq \mathcal{I}$. Given a structure $\mathcal{A}$, we use $\Sigma_{\mathcal{A}} := Dom\ \mathcal{A}$ to denote the **signature of** $\mathcal{A}$.

A type T is **licensed** by a signature Σ if $\mathcal{N}T \subseteq \Sigma$. A term t is **licensed** by a signature Σ if the following conditions hold:

1. $\mathcal{N}t - Var \subseteq \Sigma$.
2. $\forall x \in \mathcal{N}t\colon\ \mathcal{N}(\tau x) \subseteq \Sigma$

An interpretation $\mathcal{I}$ is **licensed** by a structure $\mathcal{A}$ if $\mathcal{A} \subseteq \mathcal{I}$. A type or a term are **licensed** by a structure if they are licensed by the signature of the structure.

Proposition 3.6 (Coincidence) Let $\mathcal{I}$ and $\mathcal{I}'$ be licensed by $\mathcal{A}$. Then:

1. If T is licensed by $\mathcal{A}$, then $\mathcal{I}T = \mathcal{I}'T$.
2. If t is licensed by $\mathcal{A}$ and $\mathcal{I}$ and $\mathcal{I}'$ agree on all variables in t, then $\hat{\mathcal{I}}t = \hat{\mathcal{I}}'t$.

Example 3.7 We present examples for a signature and a structure. We start with the description of a signature Σ:

$$\begin{aligned} 0, 1 &: B \\ \rightarrow &: B \rightarrow B \rightarrow B \end{aligned}$$

The described signature Σ consists of a sort B and three different constants 0, 1, and $\rightarrow$. Since we base everything on the axiomatization of terms, we cannot name concrete sorts and constants. However, we can assume that the symbol B denotes a concrete sort, and that the symbols 0, 1, and $\rightarrow$ denote concrete constants of the types specified in the description of the signature (existence guaranteed by Inf). This way we get what we want together with a nice notation. We can now define a structure $\mathcal{B}$ that interprets the names of Σ:

$$\begin{aligned} \mathcal{B}B &= \mathbb{B} \\ \mathcal{B}0 &= 0 \\ \mathcal{B}1 &= 1 \\ \mathcal{B}(\rightarrow) &= \lambda v \in \mathbb{B}.\, \lambda w \in \mathbb{B}.\, \max\{1 - v, w\} \end{aligned}$$ ∎

Two structures $\mathcal{A}$ and $\mathcal{B}$ are **isomorphic** if $\Sigma_{\mathcal{A}} = \Sigma_{\mathcal{B}}$ and for every type T licensed by $\Sigma_{\mathcal{A}}$ there exists a bijection $\gamma_T\colon \mathcal{A}T \rightarrow \mathcal{B}T$ such that:

1. For every constant $c \in \Sigma_{\mathcal{A}}$: $\gamma_{\tau c}(\mathcal{A}c) = \mathcal{B}c$.
2. For every type $T_1 \rightarrow T_2$ licensed by $\Sigma_{\mathcal{A}}$ and ever function $f \in \mathcal{A}(T_1 \rightarrow T_2)$: $(\gamma_{T_1 \rightarrow T_2})f = \{\, (\gamma_{T_1} v_1, \gamma_{T_2} v_2) \mid (v_1, v_2) \in f \,\}$.

To show that two structures $\mathcal{A}$ and $\mathcal{B}$ are isomorphic, it suffices to exhibit a bijection $\gamma_C\colon \mathcal{A}C \rightarrow \mathcal{B}C$ for every sort $C \in \Sigma_{\mathcal{A}}$. The bijections for the functional types can then obtained by recursion according to condition (2). Of course, one has to check that condition (1) is satisfied.

3.3 Equations

An **equation** of type T is a pair (s, t) of two terms $s{:}T$ and $t{:}T$. If there is no danger of confusion, we will write an equation (s, t) as $s{=}t$. An equation $s{=}t$ is

licensed by a signature Σ if s and t are licensed by Σ. We arrange the following notations:

$$\begin{array}{rcll}
e \in \mathit{Equ} & := & \{\,(s,t) \mid \tau s = \tau t\,\} & \text{equations}\\
\mathcal{N}e & := & \mathcal{N}s \cup \mathcal{N}t \quad \text{if } e = (s,t) & \text{names}\\
\mathcal{I} \models e & :\iff & \hat{\mathcal{I}}s = \hat{\mathcal{I}}t \quad \text{if } e = (s,t) & \mathcal{I}\ \textbf{satisfies}\ e\\
\mathcal{A} \models e & :\iff & \forall \mathcal{I}\colon\ \mathcal{A} \subseteq \mathcal{I} \Rightarrow \mathcal{I} \models e & \mathcal{A}\ \textbf{satisfies}\ e\\
\mathit{VE}\,\mathcal{A} & := & \{\,e \mid \mathcal{A} \models e\,\} & \textbf{valid equations}
\end{array}$$

Note that a structure $\mathcal{A}$ satisfies an equations e if and only if all interpretations licensed by $\mathcal{A}$ satisfy e. Instead of $\mathcal{A}$ satisfies e we also say that e is **valid in** $\mathcal{A}$.

For the structure $\mathcal{B}$ from Example 3.7 we have the following:

$$\begin{array}{l}
\mathcal{B} \models 1 \to x{=}x\\
\mathcal{B} \not\models 0 \to x{=}x
\end{array}$$

Proposition 3.8 Let $\mathcal{A}$ and $\mathcal{B}$ be isomorphic structures. Then $\mathcal{A} \models e \iff \mathcal{B} \models e$.

3.4 Specifications and Models

A **specification** is a set of equations. The equations of a specification are called the **axioms** of the specification. A **model** of a specification is a structure that satisfies all axioms of the specification. The **signature** of a specification is the least signature that licenses all axioms of the specification. We use Σ_A to denote the signature of a specification A and arrange the following notations:

$$\begin{array}{rcll}
A, E \subseteq \mathit{Equ} & & & \text{specifications}\\
\mathcal{A} \models A & :\iff & \forall e{\in}A\colon\ \mathcal{A} \models e & \mathcal{A}\ \text{model of}\ A,\ \mathcal{A}\ \textbf{satisfies}\ A\\
\mathcal{N}A & := & \bigcup\{\,\mathcal{N}e \mid e \in A\,\} &
\end{array}$$

Figure **??** shows the description of a specification Bool. Both the signature and the axioms are described. The explicit description of the signature provides notations for the sorts and constants of the specification. A declaration of the variables used in the axioms is not necessary since their types can be inferred from the axioms:

$$\begin{array}{l}
x : B\\
f : B \to B
\end{array}$$

Convince yourself that the structure $\mathcal{B}$ from Example 3.7 is a model of Bool.

Specification	Bool	
Sorts	B	
Constants	$0, 1 \colon B$ $\to \colon B \to B \to B$	
Axioms	$0 \to x = 1$	I0
	$1 \to x = x$	I1
	$f0 \to f1 \to fx = 1$	BCA (Boolean case analysis)

Figure 1: Specification Bool

A specification A entails an equation e semantically if every model of A satisfies e:

$$A \vDash e \;:\!\iff\; \forall \text{ model } \mathcal{A} \text{ of } A\colon\; \mathcal{A} \vDash e \qquad\qquad A \textbf{ entails } e \textbf{ semantically}$$

Our definition of models is quite liberal. In particular it admits models that interpret a sort C with a one-element set. Such models satisfy all equations of type C. In fact, every structure that interprets all sorts with one-element sets will be a model of every specification.

A **proper model** of a specification A is a model $\mathcal{A}$ of A such that $\Sigma_{\mathcal{A}} = \Sigma_A$ and $\mathcal{A}C$ has at least 2 elements for every sort $C \in \mathit{Dom}\,\mathcal{A}$.

A specification is **categorical** if it has a proper model and all its proper models are isomorphic.

Proposition 3.9 The specification Bool from Figure **??** is categorical.

Proof Let $\mathcal{A}$ be a proper model of Bool. It suffices to show that $\mathcal{A}B \subseteq \{\mathcal{A}0, \mathcal{A}1\}$ since then $\mathcal{A}B = \{\mathcal{A}0, \mathcal{A}1\}$ and $\mathcal{A}0 \neq \mathcal{A}1$ by the properness of $\mathcal{A}$ and hence $\mathcal{A}(\to)$ is determined by the axioms I0, I1.

Suppose there exists a value $v \in \mathcal{A}B - \{\mathcal{A}0, \mathcal{A}1\}$. Then there exists an interpretation $\mathcal{I}$ such that $\mathcal{A} \subseteq \mathcal{I}$ and

$$\begin{aligned} \mathcal{I}x &= v \\ \mathcal{I}fv &= v \\ \mathcal{I}f(\mathcal{A}0) &= \mathcal{A}1 \\ \mathcal{I}f(\mathcal{A}1) &= \mathcal{A}1 \end{aligned}$$

Since $\mathcal{I}$ satisfies Axiom BCA and I1, we know $\hat{\mathcal{I}}(fx) = \hat{\mathcal{I}}1$. Hence $v = \mathcal{A}1$, which contradicts our assumption. ■

We now know that the structure $\mathcal{B}$ from Example 3.7 is the only proper model of Bool, up to isomorphism. This means that the specification Bool specifies everything that is essential about $\mathcal{B}$.

We arrange the following notations:

$A \vDash A'$	$:\Longleftrightarrow \;\; \forall e \in A' \colon \; A \vDash e$	A **entails** A' **semantically**
$A \mathrel{\vDash\!\!\!\dashv} A'$	$:\Longleftrightarrow \;\; A \vDash A' \;\wedge\; A' \vDash A$	A, E **semantically equivalent**

Proposition 3.10

- $A \vDash A' \iff$ every model of A is a model of A'
- $A \mathrel{\vDash\!\!\!\dashv} A' \iff$ A and A' have the same models