3.8 Definitions Within Predicate Logic

Before using predicate logic as a working language one has to add one more facility which is indispensable for concise formalization, namely the facility of *defining new concepts* in terms of available concepts in predicate logic. In principle, definitions can be eliminated but practically they are indispensable because

- they allow to *make formulae shorter* and
- they are the means for *structuring knowledge* presented in predicate logic.

Practical mathematics would hardly be conceivable without definitions.

3.8.1 The Four Basic Types of Definitions

There are four basic types of definitions available in predicate logic:

- (explicit) definitions of predicate symbols,
- explicit definitions of functions symbols,
- implicit non-unique definitions of function symbols,
- implicit unique definitions of function symbols.

We give one example for each of the four types.

In the examples we allow various simplifications of predicate logic notation! In particular we sometimes use natural language constructs for predicate and function symbols and allow phrases like "i is reducible" instead of "is-reducible(i)" etc. Also, we omit universal quantifiers at the beginning of a formula (i.e. free variables are considered to be universally quantified).

Example 3.1 The following formula *defines the predicate symbol* "is reducible" (unary) using the binary predicate symbol $|$ ("divides"):

$$i \text{ is reducible} :\leftrightarrow \forall f (f \mid i \rightarrow (f = 1 \vee f = i)).$$

The part of the definition to the left of the colon is called the "definiendum" (Latin: "to be defined") and the right-hand part is called the "definiens" (Latin: "the defining"). The colon is *not* actually part of the definition. It is a symbol on the meta-level for indicating that this formula is a definition. In fact we will see that it is clear from the structure of the formula whether or not it is a definition.

Example 3.2 The following formula *explicitly defines the 4-ary function symbol* "determinant" using the binary function symbols − and $*$:

$$\text{determinant}(a_{1,1}, a_{1,2}, a_{2,1}, a_{2,2}) := a_{1,1} * a_{2,2} - a_{1,2} * a_{2,1}.$$

Example 3.3 The following formula *non-uniquely implicitly defines* (e.g. in the theory of complex numbers) the unary function symbol $\sqrt{\ }$ using the binary function symbols $*$:

$$\sqrt{x} := \text{a } y \text{ such that } y * y = x.$$

Here, the new quantifier "a ... such that ..." appears. Such a formula should just be conceived as an abbreviation for the following formula:

$$y = \sqrt{x} \rightarrow y * y = x.$$

Example 3.4 The following formula *uniquely implicitly defines* (e.g. in the theory of real numbers) the unary function symbol $\sqrt{\ }$ using the binary function symbols $*$ and the binary predicate symbol $\geq$.

$$\begin{aligned} \sqrt{x} := \text{the } y \text{ such that} \\ & (x \geq 0 \rightarrow (y \geq 0 \wedge y * y = x)) \\ & \wedge \\ & (x < 0 \rightarrow y = 0). \end{aligned}$$

Here, the new quantifier "the ... such that ..." appears. Such a formula should be conceived as an abbreviation for the following formula:

$$\begin{aligned} y = \sqrt{x} \leftrightarrow \\ & (x \geq 0 \rightarrow (y \geq 0 \wedge y * y = x)) \\ & \wedge \\ & (x < 0 \rightarrow y = 0. \ \square \end{aligned}$$

When introducing new predicate and function symbols by definitions, two questions arise:

- Can the definitions introduce contradictions into a theory that so far was consistent?
- Can one derive essentially new theorems that so far were not derivable in the theory?

The answer to both questions is "no". If certain rules for definitions are followed, definitions cannot introduce contradictions. Also, no "essentially" new theorems can be derived where "essentially new" means "new after elimination of the defined symbols". In fact, defined symbols can be eliminated in the sense that each formula containing defined symbols can be systematically transformed into an equvialent formula not containing these symbols. The transformed formulae can already be derived in the old theory.

The proof of these facts about definitions is not really difficult but quite lengthy. We cannot give the proofs. However, we exactly formulate the logical facts about definitions so that practical situations can be handled in a clean way.

3.8.2 Theories

As we have seen in the above examples of implicit definitions, it is important to consider the underlying theory for deciding whether or not a definition is appropriate. Hence, for making the formal properties of definitions clear we need the notion of a "theory", i.e. knowledge formulated in predicate logic. "Theories" are characterized by the language in which they are formulated, i.e. by a domain of symbols, and by their "knowledge base" or "set of axioms", i.e. by a set of formulae from which all other facts of the theory can be derived by reasoning.

Let V be a fixed set of variables.

Definition 3.5 (Theory) T is called a theory of first order predicate logic iff there exist S and F such that $T = (S, F)$ and

S is a domain of non-logical constants disjoint from V and
F is a set of formulae over V and S.

Definition 3.6 (Theorem) f is a theorem of a theory (S, F) iff $F \vdash_{ND} f$.

Definition 3.7 (Extension) The theory $T' = ((FS', RS', AR'), F')$ is an extension of the theory $T = ((FS, RS, AR), F)$ iff

$FS \subseteq FS'$,
$RS \subseteq RS'$,
for all $fs \in FS, AR(fs) = AR'(fs)$,
for all $rs \in RS, AR(rs) = AR'(rs)$,
for all f,
    if f is a theorem of T then f is a theorem of T'. □

The latter condition can also be replaced by the condition that all f in F must be theorems in T'. (However, it is not necessary that $F \subseteq F'$.)

Definition 3.8 (Conservative Extension) The theory $T' = (S', F')$ is a conservative extension of the theory $T = (S, F)$ iff T' is an extension of T and

for all formulae f over S,
    if f is a theorem of T' then f is a theorem of T . □

Let, for the next four subsections $T = (S, F)$, with $S = (FS, RS, AR)$, be an arbitrary but fixed theory.

3.8.3 Definitions of Predicate Symbols

Definition 3.9 (Form of Explicit Definitions) A formula d is a definition of the predicate symbol rs over the theory T iff d has the form

$$\forall v_1, \ldots, v_n(\ rs(v_1, \ldots, v_n) :\leftrightarrow f)$$

where rs is an n-ary relation symbol not occurring in RS and f is a formula over S in which no variables other than the distinct variables $v_1, \ldots, v_n$ are free.

Definition 3.10 (Translated Formula) Let $T' := (S', F')$ be the extension of T, where

$S' := (FS, RS \cup \{rs\}, AR')$,
$F' := F \cup \{d\}$, and
d is a definition of rs over T of the form
    $rs(v_1, \ldots, v_n) :\leftrightarrow f$.

Then, for each formula g' over S', the translation of g' is the formula g that results from g' by replacing each part $rs(t_1, \ldots, t_n)$ by $f[(v_1, \ldots, v_n) \leftarrow (t_1, \ldots, t_n)]$.

Lemma 3.11 (Elimination of Defined Symbol) With the notation of the previous definition we have:

$$F' \vdash_{ND} (g' \leftrightarrow g), \tag{1}$$
$$F' \text{ is a conservative extension of } F, \tag{2}$$
$$F' \vdash_{ND} g' \text{ iff } F \vdash_{ND} g. \tag{3}$$

□

(3) means that the defined predicate symbol rs may always be "eliminated" from any formula g' such that the resulting formula is derivable in F iff g' is derivable in F'. (3) is an easy consequence of (1) and (2).

Note that the condition on the variables is crucial in definitions of predicate symbols.

Example 3.12 The "definition"

f is a factor $:\leftrightarrow f \mid x$

introduces a contradiction (i.e. the extension of the given theory by this definition is not "conservative"), namely

3 is a factor $:\leftrightarrow 3 \mid 3$, and
3 is a factor $:\leftrightarrow 3 \mid 5$.

Hence, 3 is a factor because $3 \mid 3$, and 3 is not a factor, because not $3 \nmid 5$, a contradiction. □

In practice, the definitions of predicate symbols are sometimes also given in a form where the left-hand side does not only contain variables but terms. This must be handled with care and is only possible if the occurring terms have the properties of injective "pairing functions". In fact we used this type of definition several times on the "metalevel".

Example 3.13

(S, F) is a theory $:\leftrightarrow$
  S is a domain of symbols $\wedge$ F is a set of formulae.

"Is a theory" is a unary predicate symbol (on the metalevel), "(S, F)" is a term, not a variable!. Such a "definition" cannot introduce any contradiction because for the function symbol "$(,)$" in set theory the following property holds:

$$(x, y) = (x', y') \leftrightarrow x = x' \wedge y = y'. \qquad \text{(uniqueness)}$$

Hence S and F are uniquely determined by (S, F). Therefore it is not possible to derive the contradiction that something is a theory and is not a theory. In fact the above "definition" can always be rewritten in the following form (which, however, is clumsy):

T is a theory $:\leftrightarrow$
  $\exists S, F(T = (S, F) \wedge$ S is a domain of symbols $\wedge$
    F is a set of formulae). □

A "definition" having terms on the left-hand side is not allowed if the uniqueness property cannot be proven in the theory. Consider the following example:

Example 3.14

$x + y$ **is a nice sum** $\leftrightarrow x = 2 * y$.

In fact, this definition leads to a contradiction:

$6 + 3$ is a nice sum $\leftrightarrow 6 = 2 * 3$,
$5 + 4$ is a nice sum $\leftrightarrow 5 = 2 * 4$.

Hence, 9 is both a nice sum and not a nice sum.

3.8.4 Properties of Explicit Definitions of Function Symbols

Definition 3.15 (Form of Explicit Definitions) A formula d is a definition of the function symbol *fs* over the theory T iff d has the form

$$\forall v_1, \ldots, v_n(\mathit{fs}(v_1, \ldots, v_n) := t)$$

where *fs* is an n-ary function symbol not occurring in FS and t is a term over S in which no variables other than the distinct variables $v_1, \ldots, v_n$ are free. □

The notion of the translated formula g of a formula g' involving the new function symbol *fs* and the way the new function symbol can be eliminated is exactly analogous to the case of defined predicate symbols.

Again, the condition on the variables is crucial in the explicit definitions of function symbols.

Example 3.16 The "definition"

$$f(x) := x * y$$

introduces a contradiction. Why? □

Again, the "explicit" definitions of function symbols are sometimes also given in a form where the left-hand side does not only contain variables but terms. This is again possible if the occurring terms have the properties of injective "pairing functions".

Example 3.17
On the metalevel we could define

$$\langle (t_1 \equiv t_2) \rangle_A := \mathrm{equ}(\langle t_1 \rangle_A, \langle t_2 \rangle_A),$$
$$x = y \rightarrow \mathrm{equ}(x, y) := \mathbf{T},$$
$$\neg x = y \rightarrow \mathrm{equ}(x, y) := \mathbf{F}.$$

Here, we used $\equiv$ instead of $=$ for denoting equality on the object level in order to distinguish it from the $=$ on the metalevel. Strictly, this is not an explicit definition. However, it can easily be transformed into the following explicit definition of the binary function symbol "$\langle , \rangle$" because the components t_1 and t_2 are uniquely determined in the formula $t_1 \equiv t_2$. ("$\equiv$" is a function symbol on the metalevel!).

$$\langle f \rangle_A := \mathrm{equ}(\langle \mathrm{op}_1(f) \rangle_A, \langle \mathrm{op}_2(f) \rangle_A).$$

Here, $\equiv$ again has the essential "pairing function property"

$$(x \equiv y) = (x' \equiv y') \leftrightarrow (x = x') \wedge (y = y')$$

and op_1, op_2 are the corresponding "projection functions" satisfying

$$\mathrm{op}_1(x \equiv y) = x,$$
$$\mathrm{op}_2(x \equiv y) = y,$$
$$\mathrm{op}_1(z) \equiv \mathrm{op}_2(z) = z. \ \square$$

3.8.5 Properties of Non-Unique Implicit Definitions of Function Symbols

Definition 3.18 (Form of Definitions) A formula d is a non-unique implicit definition of the function symbol fs over the theory T iff d has the form

$$\forall v_1, \ldots, v_n, v(v = fs(v_1, \ldots, v_n) \rightarrow f)$$

where fs is an n-ary function symbol not occurring in FS and f is a formula in which no variables other than the distinct variables $v_1, \ldots, v_n, v$ are free. □

Note that d is equivalent to

$$f[v \leftarrow fs(v_1, \ldots, v_n)].$$

We should also mention that a formula d of the above kind is not always considered a "definition" of the function symbol. Some authors prefer to reserve the word "definition" for what we call here "unique implicit definitions" (see next subsection). In fact it is not possible to eliminite non-uniquely defined function symbols. Still, the expressive power of a theory is not essentially enhanced by non-uniquely defined function symbols:

Lemma 3.19 (Extension is Conservative) Let $T' := (S', F')$ be the extension of T, where ∀ If the formula $\forall v_1, \ldots, v_n \exists v\ (f)$ are Theo: (existence condition)

$S' := (FS, RS \cup \{rs\}, AR')$,
$F' := F \cup \{d\}$, and
d is a non-unique implicit definition of fs over T of the form
$\forall v_1, \ldots, v_n, v(v = fs(v_1, \ldots, v_n) \rightarrow f)$.

Then F' is a conservative extension of F. □

It is crucial that the (existence condition) is a theorem in T. Otherwise, a contradiction could be introduced by an implicit definition.

Example 3.20 Let

$$y = \sqrt{x} \rightarrow y * y = x$$

be a formula in which the variables range over the real numbers. Then

$$\sqrt{-1} * \sqrt{-1} = -1$$

and, hence,

$$\exists y(y * y = -1)$$

in contradiction to

$$\neg\exists y(y * y = -1),$$

which can be proven in the theory of real numbers. □

3.8.6 Properties of Unique Implicit Definitions of Function Symbols

Definition 3.21 (Form of Definitions) A formula d is a unique implicit definition of the function symbol fs over the theory T iff d has the form

$$\forall v_1, \ldots, v_n, v(v = fs(v_1, \ldots, v_n) \leftrightarrow f)$$

where fs is an n-ary function symbol not occurring in FS and f is a formula in which no variables other than the distinct variables $v_1, \ldots, v_n, v$ are free.

Definition 3.22 (Translated Formula) Let $T' := (S', F')$ be the extension of T, where

$S' := (FS, RS \cup \{fs\}, AR')$,
$F' := F \cup \{d\}$, and
d is a unique implicit definition of fs over T of the form
$\forall v_1, \ldots, v_n, v(v = fs(v_1, \ldots, v_n) \leftrightarrow f)$.

Then, for each atomic formula g' over S', the translation of g' is the formula g that results from g' by recursive application of the following step: If fs does not occur in g' then $g := g'$ otherwise g' can be written in the form

$$h[w \leftarrow fs(t_1, \ldots, t_n)],$$

where h is an atomic formula, w is a variable and fs does not occur in any of the $t_1, \ldots, t_n$.Then g is

$$\exists w(f[(v_1, \ldots, v_n, v) \leftarrow (t_1, \ldots, t_n, w)] \wedge h^*),$$

where h^* is the translation of h. (Note that h contains one less occurrence of *fs* than g'.)

The translation of non-atomic formulae proceeds by translating each atomic part.

Lemma 3.23 (Elimination of Defined Symbol) With the notation of the previous definition we have: If the formulae

$$\forall v_1, \ldots, v_n \exists v(f), \text{ and} \qquad \text{(existence condition)}$$
$$\forall v_1, \ldots, v_n, v, v'(f \wedge f[v \leftarrow v'] \rightarrow v = v'), \qquad \text{(uniqueness condition)}$$

are theorems of T then

$$F' \vdash_{ND} (g' \leftrightarrow g), \qquad (1)$$
$$F' \text{ is a conservative extension of } F, \qquad (2)$$
$$F' \vdash_{ND} g' \text{ iff } F \vdash_{ND} g. \qquad (3)$$

□

It is crucial that both the existence condition and the uniqueness condition is a theorem in T. Otherwise, a contradiction could be introduced by an implicit definition.

Example 3.24 Let

$$y = \sqrt{x} :\leftrightarrow$$
$$(x \geq 0 \rightarrow y * y = x)$$
$$\wedge$$
$$(x < 0 \rightarrow y = 0)$$

be a formula in which the variables range over the real numbers. The uniqueness condition is not satisfied. We obtain the following contradiction:

$$3 = \sqrt{9} \leftrightarrow$$
$$(9 \geq 0 \rightarrow 3 * 3 = 9)$$
$$\wedge$$
$$(9 < 0 \rightarrow 3 = 0)\,, \qquad (1)$$
$$-3 = \sqrt{9} \leftrightarrow$$
$$(9 \geq 0 \rightarrow (-3) * (-3) = 9)$$
$$\wedge$$
$$(9 < 0 \rightarrow -3 = 0)\,, \qquad (2)$$
$$3 = \sqrt{9}, \qquad \text{from (1)}$$
$$-3 = \sqrt{9}, \qquad \text{from (2)}$$
$$3 = -3. \ \square$$

Note that for definitions by "cases" in which each case has the form of an explicit definition, both the (existence condition) and the (uniqueness condition) are automatically satisfied.

Example 3.25

$$x = y \rightarrow \text{equ}(x, y) = \mathbf{T},$$
$$\neg(x = y) \rightarrow \text{equ}(x, y) = \mathbf{F},$$

is such a definition. Usually, these definitions are written in the form

$$\text{equ}(x, y) = \begin{cases} \mathbf{T}, & \text{if } x = y, \\ \mathbf{F}, & \text{otherwise.} \end{cases} \quad \square$$

Finally, we give an example of the above translation process.

Example 3.26 (Translation of a Formula) Let d be the following unique implicit definition of the unary function symbol "reverse" (where the variable f ranges over sequences, for simplicity of fixed length 10, and i and j range over integers):

$$f' = \text{reverse}(f) :\leftrightarrow \forall i(1 \leq i \leq 10 \rightarrow f_i = f'_{11-i}).$$

Let g be the following formula:

$$\text{reverse}(\ \text{reverse}(f)) = f.$$

First step of translation:

$$\exists f'(f' = \text{reverse}(f) \wedge \text{reverse}(f') = f),$$
$$\exists f'(\forall i(1 \leq i \leq 10 \rightarrow f_i = f'_{11-i})$$
$$\quad \wedge \text{reverse}(f') = f).$$

Second step of translation:

$$\exists f'\ (\forall i(1 \leq i \leq 10 \rightarrow f_i = f'_{11-i})$$
$$\quad \wedge \exists f''(f'' = \text{reverse}(f') \wedge f'' = f));$$
$$\exists f'(\forall i(1 \leq i \leq 10 \rightarrow f_i = f'_{11-i})$$
$$\quad \wedge \exists f''(\forall i(1 \leq i \leq 10 \rightarrow f'_i = f''_{11-i})$$
$$\quad\quad \wedge f'' = f)).$$